Network Adiabatic Theorem: An Efficient Randomized Protocol for Contention Resolution

Shreevatsa Rajagopalan
Massachusetts Institute of Technology
vatsa@mit.edu

Devavrat Shah
Massachusetts Institute of Technology
devavrat@mit.edu

Jinwoo Shin*
Massachusetts Institute of Technology
jinwoos@mit.edu

ABSTRACT

The popularity of *Aloha*(-like) algorithms for resolution of contention between multiple entities accessing common resources is due to their extreme simplicity and distributed nature. Example applications of such algorithms include Ethernet and recently emerging wireless multi-access networks. Despite a long and exciting history of more than four decades, the question of designing an algorithm that is *essentially* as simple and distributed as Aloha while being efficient has remained unresolved.

In this paper, we resolve this question successfully for a network of queues where contention is modeled through independent-set constraints over the network graph. The work by Tassiulas and Ephremides (1992) suggests that an algorithm that schedules queues so that the summation of "weight" of scheduled queues is maximized, subject to constraints, is efficient. However, implementing such an algorithm using Aloha-like mechanism has remained a mystery. We design such an algorithm building upon a Metropolis-Hastings sampling mechanism along with selection of "weight" as an appropriate function of the queue-size. The key ingredient in establishing the efficiency of the algorithm is a novel *adiabatic*-like theorem for the underlying queueing network, which may be of general interest in the context of dynamical systems.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Stochastic processes, Markov processes, Queueing theory; C.2.1 [**Network Architecture and Design**]: Distributed networks, Wireless communication

General Terms

Algorithms, Performance, Design

Keywords

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

1. INTRODUCTION

A multiple-access channel is a broadcast channel that allows multiple users to communicate with each other by sending messages onto the channel. If two or more users simultaneously send messages, then the messages interfere with each other (collide), and the messages are not transmitted successfully. The channel is not centrally controlled. Instead, users need to use a distributed protocol or algorithm to resolve contention. The popular Aloha protocol or algorithm was developed more than four decades ago to address this (e.g. see [1]). The key behind such a protocol is using collision or busyness of the channel as a signal of congestion and then reacting to it using a simple randomized rule.

Although the most familiar multiple-access channels are wireless multiple-access media (*a la* IEEE 802.11 standards) and wired local-area networks (such as Ethernet networks), now multiple-access channels are also being implemented using a variety of technologies including packet-radio, fiber-optics, free-space optics and satellite transmission (e.g. see [12]). These multiple-access channels are used for communication in many distributed networked systems, including emerging communication networks such as the wireless *mesh* networks [25].

Despite the long history and great importance of multi-access contention-resolution protocols, the question of designing an efficient Aloha-like simple protocol (algorithm) has remained unresolved in complete generality even for one multiple-access channel. In this paper, we are interested in designing a distributed contention resolution protocol for a *network* of multiple-access channels in which various subsets of these network users (nodes) interfere with each other. For example, in a wireless network placed in a geographic area, two users interfere with each other if they are nearby and do not interfere if they are far apart. Such networks can be naturally modeled as queueing networks with contentions modeled through independent-set constraints over the network interference graph. For this setup, we will design a simple randomized, Aloha-like, algorithm that is efficient. Indeed, as a special case, it resolves the classical multiple-access single broadcast channel problem as well.

*Author names appear in the alphabetical order of their last names. All authors are with Laboratory for Information and Decision Systems, MIT. This work was supported in parts by NSF projects HSD 0729361, CNS 0546590, TF 0728554 and DARPA ITMANET project.

SIGMETRICS/Performance'09, June 15–19, 2009, Seattle, WA, USA.

1.1 Related work

Design and analysis of multiple-access contention resolution algorithms have been of great interest for four decades across research communities. Due to its long and rich history, it will be impossible for us to provide a complete history. We will describe a few of these results that are closer to our result. Primarily, research has been divided into two classes: single channel multiple-access protocols and network multiple-access protocols.

Single multi-access channel. The research in single channel setup evolved into two branches: (a) *Queue-free* model and (b) *Queueing* model. For the queue-free model, some notable works about inefficiency of certain class of protocols are due to Kelly and McPhee [18][19][20], Aldous [2], Goldberg, Jerrum, Kannan and Paterson [10] and Tsybakov and Likhanov [32] — the last one establishing impossibility of throughput optimality for any protocol in the queue-free model. On the positive side for the queue-free model, work by Mosley and Humblet establishes existence of a "tree-protocol" with a positive rate. There are many other results on related models; we refer an interested reader to Ephremides and Hajek [6] and the online survey by Leslie Goldberg [11].

For the queueing model, a notable positive result is due to Hastad, Leighton and Rogoff [15] that establishes that if there are N users with each having the same rate λ/N, a (polynomial) version of the standard back-off protocol is stable as long as $\lambda < 1$. Of course, this does not extend to case when users have different rates even though their net rate might be less than 1. In summary, there is no known algorithm that operates without any information exchange between queues while being efficient (or throughput-optimal) in the queueing model even for single multi-access channel.

Network of multiple-access channels. The lack of any efficient protocol without information exchange even for a single channel has led to an exciting progress in the past 5 years or so for designing *message-passing* algorithms for a network of multiple-access channels. Interest in such algorithms has been fueled by emergence of wireless multi-hop networks as a canonical architecture for an access network in a residential area or a metro-area network in a dense city. In what follows, we briefly describe some of the key recent results.

Primarily, the focus has been on a network queueing model with an associated interference graph. Here two queues can not transmit simultaneously if they are neighbors in their interference graph. Therefore effectively a contention-resolution protocol or scheduling algorithm is required to schedule, at each time, transmissions of queues that form an *independent set* of the network interference graph (see Section 2 for a detailed formal description).

Now, ignoring implementation concerns, the work by Tassiulas and Ephremides [31] established that the maximum weight (MW) algorithm, which schedules queues satisfying independent-set constraints with maximum summation of their weights, where the weight of a queue is its queue-size, is throughput-optimal. However, implementing the MW algorithm, i.e. finding a maximum weighted independent set in the network interference graph in a distributed and simple manner, is a daunting task. Ideally, one wishes to design a MW algorithm that is as simple as the random-access protocols (or Aloha). This has led researchers to exploit two approaches: (1) design of random-access algorithms with access probabilities that are arrival-rate-aware, and (2) design of distributed implementations of MW algorithms.

We begin with the first line of approach. Here the question boils down to finding appropriate channel access probabilities for head-of-line packets as a function of their local history (i.e. age, queue-size or backoff). In a very important and exciting recent work, Bordenave, McDonald and Proutiére [3] obtained characterization of the capacity region of a multi-access network with given (fixed) access probabilities in the limit of the large network size (mean-field limit). Notably, this work settled an important question that had remained open for a while. On the flip side, it provides an approximate characterization of the capacity region for a small network (precise approximation error is not clear to us). Also, a fixed set of access probabilities is unlikely to work for any arrival rate vector in the capacity region. Therefore, to be able to support a larger capacity region, one needs to select access probabilities that should be adjusted depending on system arrival process and this will require some information exchange.

In an earlier work motivated by this concern, Marbach [21] as well as Eryilmaz, Marbach and Ozdaglar [22] did consider the selection of access probabilities based on the arrival rates. In a certain asymptotic sense, they established that their rate-aware selection of the access probabilities allocate rates to queues so that the allocated rates are no less than the arrival rates. A caveat of their approach was "saturated system" analysis and the goodness of the algorithm in an asymptotic sense.

Another sequence of works by Gupta and Stolyar [14], Stolyar [30] and Liu and Stolyar [17] considered random-access algorithms where the access probabilities are determined as a function of the queue-sizes by means of solving an optimization problem in a distributed manner. The algorithm has certain throughput (Pareto) optimality property. However, it requires solving an optimization problem in a distributed manner every time! This can lead to a *lot* of information exchange per time step. We take note of a very recent work by Jiang and Walrand [16] that employs a similar approach for determining the access probabilities using arrival rate information. They also speculate an intuitively pleasing connection between their rate-aware approach with a queue-aware approach. However, they *do not* establish the stability of the network under their algorithm. Interestingly enough, we strongly believe that our proof techniques may establish the stability of (a variant of) their algorithm.

Many of these approaches for determining access probabilities based on rates are inherently not 'robust' against change of rates and this is what strongly motivates queue-based approaches, i.e. distributed implementation of MW algorithm. As the first non-trivial step, Modiano, Shah and Zussman [24] provided a totally distributed, simple *gossip* algorithm to find an approximate MW schedule each time for matching constraints (it naturally extends to independent-set constraints and to cross-layer optimal control of a multi-hop network, e.g. see [7]). This algorithm is throughput-optimal, like the standard MW algorithm it does not require information about arrival rates, or it does not suffer from the caveat of "saturated system" analysis. In this algorithm, the computation of each schedule requires up to $O(n^3)$ information exchange. In that sense, the algorithm is not *implementable* and merely a proof-of-concept. Mo-

tivated by this, Sanghavi, Bui and Srikant [26] designed (almost) throughput-optimal algorithm with constant (but large) amount of information exchange per node for computing a new schedule. However, their approach is applicable only to *matching* constraints and it does require (large) constant amount of co-ordination between local neighborhoods for good approximation guarantee (e.g. for 95% throughput, it requires co-ordination of neighbors within $\sim$ 20 hops!). Finally, their approach does not extend to independent-set constraints.

In summary, none of the random-access based algorithms that are studied in the literature have desirable properties, as one or more of the following limitation exists. (1) They assume "saturated system", hence need to solve an optimization problem using the knowledge of arrival rate that requires a lot of message-passing. (2) The capacity region is not the largest possible. (3) The distributed implementation of the MW algorithm, though provides the proof-of-concept of existence of a distributed, simple and throughput-optimal algorithm; they require a lot of information exchange for the computation of each schedule. That is, they are not simple or elegant enough (like Aloha) to be of practical utility.

1.2 Contributions

As the main contribution of this paper, we design a throughput-optimal and stable[1] random-access algorithm for a network of queues where contention is modeled through independent set constraints. Our random-access algorithm is elegant, simple and, in our opinion, of great practical importance. And it indeed achieves the desired throughput-optimality property by making the random-access probabilities time-varying and a function of the queue-size. The key to the efficiency of our algorithm lies in the careful selection of this function.

To this end, first we observe that if queue-sizes were *fixed* then one can use Metropolis-Hastings based sampling mechanism to sample independent sets so that sampled independent sets provides a good approximation of the MW algorithm. As explained later in detail (or an informed reader may gather from the literature), the Metropolis-Hastings based sampling mechanism is essentially a continuous time random access protocol (like Aloha). Therefore, for our purposes the use of Metropolis-Hastings sampler would suffice only if queue-sizes were *fixed*. But queue-sizes change essentially at unit rate and the time for Metropolis-Hastings to reach "equilibrium" can be much longer. Therefore, in essence the Metropolis-Hastings mechanism may never reach "equilibrium" and hence such an algorithm may perform very poorly.

We make the following crucial observations to resolve this issue: (1) the queue-size may change at unit rate, but a function (say f) of the queue-size may change slowly (i.e. has a small derivative f'); (2) the MW algorithm is stable even when the weight is not the queue-size but some slowly changing function of the queue-size. In this paper, we will use a function $f(x) \sim \log\log x$ for this purpose. Motivated by this, we design Metropolis-Hastings sampling mechanism to sample independent sets with weights defined as this slowly changing function of the queue-size. This is likely to allow our network to be in a state so that the random-access algorithm based on Metropolis-Hastings method is essentially sampling independent sets as per the "correct" distribution all the time. As the key technical contribution, we indeed establish this non-trivial desirable result. This technical result is a "robust probabilistic" analogue of the standard adiabatic theorem [4, 13] in physics which states that *if a system changes in a reversible manner at an infinitesimally small rate, then it always remains in its ground state* (see statement of Lemma 12 and Section 5.5 for precise details).

As a consequence of this (after overcoming necessary technical difficulties), we obtain a random-access based algorithm under which the network Markov process is positive Harris recurrent (or stable) and throughput-optimal. We present simulation results to support its practical relevance. Our results (both in simulation and theory) suggest that our choice of f is critical since the natural choice of weight as the queue-size (i.e. $f(x) = x$) will not lead to a throughput-optimal algorithm.

[1]In this paper, the notion of stability is defined as positive recurrence or positive Harris recurrence of the network Markov process.

2. PRELIMINARIES

Notation. We will reserve bold letters for vectors: e.g. $\mathbf{u} = [u_i]_{i=1}^d$ denotes a d-dimensional vector; $\mathbf{1}$ and $\mathbf{0}$ denote the vector of all 1s and all 0s. Given a function $\phi : \mathbb{R} \to \mathbb{R}$, by $\phi(\mathbf{u})$ we mean $\phi(\mathbf{u}) = [\phi(u_i)]$. For any vector $\mathbf{u} = [u_i]$, define $u_{\max} = \max_i u_i$ and $u_{\min} = \min_i u_i$. For a probability vector $\pi \in \mathbb{R}_+^d$ on d elements, we will use a notation $\pi = [\pi(i)]$ where $\pi(i)$ is the probability of $i, 1 \le i \le d$.

Network model. Our network is a collection of n queues. Each queue has a dedicated exogenous arrival process through which new work arrives in the form of unit-sized packets. Each queue can be potentially serviced at unit rate, resulting in departures of packets from it upon completion of their unit service requirement. The network will be assumed to be *single-hop*, i.e. once work leaves a queue, it leaves the network. At first glance, this appears to be a strong limitation. However, as we discuss later in Section 3, the results of this paper, in terms of algorithm design and analysis, naturally extend to the case of the multi-hop setting.

Let $t \in \mathbb{R}_+$ denote the (continuous) time and $\tau = \lfloor t \rfloor \in \mathbb{N}$ denote the corresponding discrete time slot. Let $Q_i(t) \in \mathbb{R}_+$ be the amount of work in the ith queue at time t. Queues are served in First-Come-First-Serve manner. $Q_i(t)$ is the number of packets in queue i at time t, e.g. $Q_i(t) = 2.7$ means head-of-line packet has received 0.3 unit of service and 2 packets are waiting behind it. Also, define $Q_i(\tau) = Q_i(\tau^+)$ for $\tau \in \mathbb{N}$. Let $\mathbf{Q}(t)$, $\mathbf{Q}(\tau)$ denote the vector of queue-sizes $[Q_i(t)]_{1\le i\le n}$, $[Q_i(\tau)]_{1\le i\le n}$ respectively. Initially, time $t = \tau = 0$ and the system starts empty, i.e. $\mathbf{Q}(0) = \mathbf{0}$.

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

The queues are offered service as per a continuous-time (or asynchronous/non-slotted) scheduling algorithm. Each of the n queues is associated with a wireless transmission-capable device. Under any reasonable model of communica-

tion deployed in practice (e.g. 802.11 standards), in essence if two devices are close to each other and share a common frequency to transmit at the same time, there will be interference and data is likely to be lost. If the devices are far away, they may be able to simultaneously transmit with no interference. Thus the scheduling constraint here is that no two devices that might interfere with each other can transmit at the same time. This can be naturally modeled as an *independent-set* constraint on a graph (called the *interference graph*), whose vertices correspond to the devices, and where two vertices share an edge if and only if the corresponding devices would interfere when simultaneously transmitting. Specifically, let $G = (V, E)$ denote the network interference graph with $V = \{1, \ldots, n\}$ representing n nodes and

$$E = \{(i, j) : i \text{ and } j \text{ interfere with each other}\}.$$

Let $\mathcal{N}(i) = \{j \in V : (i, j) \in E\}$ denote the neighbors of node i. We assume that if node i is transmitting, then all of its neighbors in $\mathcal{N}(i)$ can "listen" to it. Let $\mathcal{I}(G)$ denote the set of all independent sets of G, i.e. subsets of V so that no two neighbors are adjacent to each other. Formally,

$$\mathcal{I}(G) = \{\boldsymbol{\sigma} = [\sigma_i] \in \{0, 1\}^n : \sigma_i + \sigma_j \leq 1 \text{ for all } (i, j) \in E\}.$$

Under this setup, the set of feasible schedules $\mathcal{S} = \mathcal{I}(G)$.

Given this, let $\boldsymbol{\sigma}(t) = [\boldsymbol{\sigma}_i(t)]$ denote the collective scheduling decision at time $t \in \mathbb{R}_+$, with $\boldsymbol{\sigma}_i(t)$ being the rate at which node i is transmitting. Then as discussed, it must be that $\boldsymbol{\sigma}(t) \in \mathcal{I}(G)$, $\boldsymbol{\sigma}_i(t) \in \{0, 1\}$ for all i, t.

The queueing dynamics induced under the above described model can be summarized by the following equation: for any $0 \leq s < t$ and $1 \leq i \leq n$,

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

where $A_i(s, t)$ denotes the cumulative arrival to queue i in time interval $[s, t]$ and $\mathbf{1}_{\{x\}}$ denotes the indicator function. Finally, define the cumulative departure process $\boldsymbol{D}(t) = [D_i(t)]$, where

$$D_i(t) = \int_0^t \sigma_i(y) \mathbf{1}_{\{Q_i(y) > 0\}} dy.$$

Performance metric. We need an algorithm to select schedule $\boldsymbol{\sigma}(t) \in \mathcal{S} = \mathcal{I}(G)$ for all $t \in \mathbb{R}_+$. Thus, a scheduling algorithm is equivalent to scheduling choices $\boldsymbol{\sigma}(t), t \in \mathbb{R}_+$. From the perspective of network performance, we would like the scheduling algorithm to be such that the queues in network remain as small as possible given the arrival process. From the implementation perspective, we wish that the algorithm be simple and distributed, i.e. perform constant number of logical operations at each node (or queue) per unit time, utilize information only available locally at the node or obtained through a neighbor and maintain as little data structure as possible at each node.

First, we formalize the notion of performance. In the setup described above, we define capacity region $\mathcal{C} \subset [0, 1]^n$ as the convex hull of the feasible scheduling set $\mathcal{I}(G) = \mathcal{S}$, i.e.

$$\mathcal{C} = \left\{ \sum_{\boldsymbol{\sigma} \in \mathcal{S}} \alpha_{\boldsymbol{\sigma}} \boldsymbol{\sigma} : \sum_{\boldsymbol{\sigma} \in \mathcal{S}} \alpha_{\boldsymbol{\sigma}} = 1 \text{ and } \alpha_{\boldsymbol{\sigma}} \geq 0 \text{ for all } \boldsymbol{\sigma} \in \mathcal{I}(G) \right\}.$$

The intuition behind this definition of capacity region comes from the fact that any algorithm has to choose schedule from $\mathcal{I}(G)$ each time and hence the time average of the 'service rate' induced by any algorithm must belong to $\mathcal{C}$. Therefore, if arrival rates $\boldsymbol{\lambda}$ can be 'served' by any algorithm then it must belong to $\mathcal{C}$. Motivated by this, we call an arrival rate vector $\boldsymbol{\lambda}$ admissible if $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}$, where

$$\boldsymbol{\Lambda} = \{\boldsymbol{\lambda} \in \mathbb{R}_+^n : \boldsymbol{\lambda} \leq \boldsymbol{\sigma} \text{ componentwise, for some } \boldsymbol{\sigma} \in \mathcal{C}\}.$$

We say that an arrival rate vector $\boldsymbol{\lambda}$ is strictly admissible if $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^o$, where $\boldsymbol{\Lambda}^o$ is the interior of $\boldsymbol{\Lambda}$ formally defined as

$$\boldsymbol{\Lambda}^o = \{\boldsymbol{\lambda} \in \mathbb{R}_+^n : \boldsymbol{\lambda} < \boldsymbol{\sigma} \text{ componentwise, for some } \boldsymbol{\sigma} \in \mathcal{C}\}.$$

Equivalently, we may say that the network is *underloaded*. Now we are ready to define the performance metric for a scheduling algorithm.

DEFINITION 1 (THROUGHPUT-OPTIMAL). *We call a scheduling algorithm* throughput-optimal*, or* stable*, or providing* 100% throughput*, if for any $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^o$ the underlying network Markov process is* positive Harris recurrent*.*

Positive Harris recurrence & its implications. For completeness, we define the well known notion of positive Harris recurrence (e.g. see [5]). We also state its useful implications to explain its desirability. In this paper, we will be concerned with discrete-time, time-homogeneous Markov process or chain evolving over a complete, separable metric space X. Let $\mathcal{B}_{\mathsf{X}}$ denote the Borel σ-algebra on X. Let $X(\tau)$ denote the state of Markov chain at time $\tau \in \mathbb{N}$.

Consider any $A \in \mathcal{B}_{\mathsf{X}}$. Define stopping time $T_A = \inf\{\tau \geq 1 : X(\tau) \in A\}$. Then the set A is called Harris recurrent if

$$\Pr_x(T_A < \infty) = 1 \qquad \text{for any } x \in \mathsf{X},$$

where $\Pr_x(\cdot) \equiv \Pr(\cdot | X(0) = x)$. A Markov chain is called Harris recurrent if there exists a σ-finite measure μ on $(\mathsf{X}, \mathcal{B}_{\mathsf{X}})$ such that whenever $\mu(A) > 0$ for $A \in \mathcal{B}_{\mathsf{X}}$, A is Harris recurrent. It is well known that if X is Harris recurrent then an essentially unique invariant measure exists (e.g. see Getoor [9]). If the invariant measure is finite, then it may be normalized to obtain a unique invariant probability measure (or stationary probability distribution); in this case X is called positive Harris recurrent.

A popular algorithm. In this paper, our interest is in scheduling algorithms that utilize the network state, i.e. the queue-size $\mathbf{Q}(t)$, to obtain a schedule. An important class of scheduling algorithms with throughput-optimality property is the well known *maximum-weight* scheduling algorithm which was first proposed by Tassiulas and Ephremides [31]. We describe the slotted-time version of this algorithm. In this version, the algorithm changes decision in the beginning of every time slot using $\mathbf{Q}(\tau) = \mathbf{Q}(\tau^+)$. Specifically, the scheduling decision $\boldsymbol{\sigma}(\tau)$ remains the same for the entire time slot τ, i.e. $\boldsymbol{\sigma}(t) = \boldsymbol{\sigma}(\tau)$ for $t \in (\tau, \tau + 1]$, and it satisfies

$$\boldsymbol{\sigma}(\tau) \in \arg\max_{\boldsymbol{\rho} \in \mathcal{S}} \sum_i \rho_i Q_i(\tau).$$

Thus, this maximum weight or MW algorithm chooses schedule $\boldsymbol{\sigma} \in \mathcal{S}$ that has the maximum weight, where weight is defined as $\boldsymbol{\sigma} \cdot \mathbf{Q}(\tau) = \sum_{i=1}^n \boldsymbol{\sigma}_i Q_i(\tau)$. A natural generalization of MW algorithm uses a weight $f(Q_i(\cdot))$ instead of $Q_i(\cdot)$ as above for some function f (e.g. see [27, 28]).

3. MAIN RESULT

This section presents the main result of this paper, namely an efficient distributed scheduling algorithm. In what follows, we begin by describing the algorithm. Our algorithm is designed with the aim of approximating the maximum weight in a distributed manner. For our distributed algorithm to be efficient (or throughput-optimal), the approximation quality of the maximum weight has to be good. As we shall establish, such is the case when the selection of weight function is done carefully. Therefore, first we describe the algorithm for a generic weight function. Next, we formally state the efficiency of the algorithm for a specific weight function. This is followed by some details for distributed implementation. Finally, we discuss the extension of the algorithm for the multi-hop setting, as well as a conjecture.

3.1 Algorithm description

As before, let $t \in \mathbb{R}_+$ denote the time. Let $\boldsymbol{W}(t) = [W_i(t)] \in \mathbb{R}_+^n$ denote the vector of weights at the n queues at time t. As we shall see, $\boldsymbol{W}(t)$ will be a certain function of the queue-sizes $\mathbf{Q}(t)$. The algorithm we describe is a continuous time algorithm that wishes to compute schedule $\boldsymbol{\sigma}(t) \in \mathcal{I}(G)$ in a distributed manner so as to have weight $\sum_i \sigma_i(t) W_i(t)$ as large as possible.

The algorithm is randomized and asynchronous. Each node has an independent Exponential clock of rate 1. Let T_k^i be the time when the clock of node i ticks for the kth time. Initially, $k = 0$ and $T_0^i = 0$ for all i. Then $T_{k+1}^i - T_k^i$ are i.i.d. and have Exponential distribution of mean 1. The nodes change their scheduling decisions only upon their clock ticks. That is, $\sigma_i(t)$ remains constant for $t \in (T_k^i, T_{k+1}^i]$. Note that due to the property of continuous random variables, no two clock ticks at different nodes will happen at the same time (with probability 1).

Let the algorithm start with null-schedule, i.e. $\boldsymbol{\sigma}(0) = [0] \in \mathcal{I}(G)$. Consider time T_k^i, the kth clock tick of node i for $k > 0$. Now node i at this particular time instant $t = T_k^i$ "listens" to the medium and does the following:

- If any neighbor is transmitting, then $\sigma_i(t^+) = 0$.
- Else, $\sigma_i(t^+) = 1$ with probability $\frac{\exp(W_i(t))}{1+\exp(W_i(t))}$ and $\sigma_i(t^+) = 0$ otherwise. This randomized decision is done independently of everything else.

We assume that if $\sigma_i(t) = 1$, then node i will always transmit data irrespective of the value of $Q_i(t)$ so that the neighbors of node i, i.e. nodes in $\mathcal{N}(i)$, can infer $\sigma_i(t)$ by "listening" to the medium.

3.2 Efficiency of algorithm

We describe a specific choice of weight $\boldsymbol{W}(t)$ for which the above described algorithm is throughput-optimal for any network graph G. In what follows, let $f(\cdot) : \mathbb{R}_+ \to \mathbb{R}_+$ be a strictly concave monotonically increasing function with $f(0) = 0$. We will be interested in functions growing much slower than $\log(\cdot)$ function. Specifically, we will use the function $f(x) = \log\log(x + e)$ in our algorithm, where $\log(\cdot)$ is the natural logarithm. For defining the weight, we will utilize a given small constant $\varepsilon > 0$. Let $\widetilde{Q}_{\max,i}(t)$ be an estimation of $Q_{\max}(t)$ at node i at time t. A straightforward algorithm to compute $\widetilde{Q}_{\max}(t)$ is described in Section 3.3. As will be established in Lemma 2, $Q_{\max}(t) - 2n \leq \widetilde{Q}_{\max,i}(t) \leq Q_{\max}(t)$ for all i and $t > 0$. Now define the weight at node i,

$$W_i(t) = \max\left\{ f(Q_i(\lfloor t \rfloor)), \frac{\varepsilon}{n} f(\widetilde{Q}_{\max,i}(\lfloor t \rfloor)) \right\}. \qquad (1)$$

For such a choice of weight, we state the following throughput optimality property of the algorithm.

THEOREM 1. *Consider any $\varepsilon > 0$. Suppose the algorithm uses weight as defined in (1) with $f(x) = \log\log(x+e)$, and $|\widetilde{Q}_{\max,i}(t) - Q_{\max}(t)|$ is uniformly bounded[2] by a constant for all t. Then, for any $\boldsymbol{\lambda} \in (1-2\varepsilon)\boldsymbol{\Lambda}^o$, the (appropriately defined) network Markov process is positive Harris recurrent.*

3.3 Distributed implementation

The goal here is to design an algorithm that is truly distributed and simple. That is, each node makes only *constant* number of operations locally each time, communicates only *constant* amount of information to its neighbors, maintains only *constant* amount of data structure and utilizes only local information. Further, we wish to avoid algorithms that satisfy the above properties by collecting some information over time. In essence, we want simple "Markovian" algorithms.

The algorithm described above, given the knowledge of node weight $W_i(\cdot)$ at node i for all i, does have these properties. Now the weight $W_i(\cdot)$ as defined in (1) depends on $Q_i(\cdot)$ and $Q_{\max}(\cdot)$ (or its estimate $\widetilde{Q}_{\max,i}(\cdot)$). Trivially, the $Q_i(\cdot)$ is known at each node. However, the computation of $Q_{\max}(\cdot)$ requires global information. Next, we describe a simple scheme in which each node maintains an estimate $\widetilde{Q}_{\max,i}(\cdot)$ at node i. To keep this estimate updated, each node broadcasts exactly one number to all of its neighbors every time slot. And, using the information received from its neighbors each time, it updates its estimate. Before describing it, we make a note of the following: In section 3.5, we provide a conjecture (supported by simulation results, see section 6) that the algorithm without the term corresponding to $\widetilde{Q}_{\max,i}(t)$ in (1) should be throughput-optimal. Therefore, for the practioner we recommend the algorithm that is conjectured in section 3.5.

Now, we state the precise procedure to compute $\widetilde{Q}_{\max,i}(t)$, the estimate of $Q_{\max}(t)$ at node i at time t. It is updated once every time slot. That is, $\widetilde{Q}_{\max,i}(t) = \widetilde{Q}_{\max,i}(\lfloor t \rfloor)$. Let $\widetilde{Q}_{\max,i}(\tau)$ be the estimate of node i at time slot $\tau \in \mathbb{N}$. Then node i broadcasts this estimate to its neighbors at the end of time slot τ. Let $\widetilde{Q}_{\max,j}(\tau)$ for $j \in \mathcal{N}(i)$ be the estimates received by node i at the end of time slot τ. Then, update

$$\widetilde{Q}_{\max,i}(\tau+1) = \max\left\{ \max_{j \in \mathcal{N}(i) \cup \{i\}} \widetilde{Q}_{\max,j}(\tau) - 1, Q_i(\tau+1) \right\}.$$

We state the following property of this estimation algorithm, the proof follows in a straightforward manner from the fact that $Q_i(\tau)$ is 1-Lipschitz.

LEMMA 2. *Assuming that graph G is connected, we have, for all $\tau \geq 0$ and all i,*

$$Q_{\max}(\tau) - 2n \leq \widetilde{Q}_{\max,i}(\tau) \leq Q_{\max}(\tau).$$

[2]See Lemma 2.

3.4 Extensions

The algorithm described here is for the single-hop network with the exogenous arrival process. As the reader will find, the key reason behind the efficiency of the algorithm is similar to the reason behind the efficiency of the standard maximum weight scheduling (here, the weight is $\log\log(\cdot)$ function of the queue-size). The standard maximum weight algorithm has a known version for a general multi-hop network with choice of routing by Tassiulas and Ephremides [31]. This is popularly known as *back pressure* algorithm, where weight of an action of transferring a packet from node i to node j is determined in terms of the difference of queue-sizes at node i and node j. Analogously, our algorithm can be modified for such a setup by using the weight of an action of transferring a packet from node i to node j as the difference of $\log\log(\cdot)$ of queue-sizes at node i and node j. The corresponding changes in algorithm described in Section 3.1 is strongly believed to be efficient using the similar proof method as that in this paper. More generally, there have been clever utilizations of such a back-pressure approach in designing congestion control and scheduling algorithm in a multi-hop wireless network, e.g. see the survey by Shakkottai and Srikant [29]. Again, we strongly believe that the utilization of our algorithm with appropriate weights will lead to a complete solution for congestion control and scheduling in a multi-hop wireless network.

3.5 A conjecture

The algorithm described for the single hop network utilizes the weight $W_i(t)$ defined as (1). This weight $W_i(t)$ depends on $Q_i(\lfloor t \rfloor)$, the queue size of node i; and $\widetilde{Q}_{\max,i}(t)$, the estimate of $Q_{\max}(t)$. Among these, the use of $\widetilde{Q}_{\max,i}(t)$ is for 'technical' reasons. While the algorithm described here provides a provably random access algorithm, we conjecture that the algorithm that operates without the use of $\widetilde{Q}_{\max,i}(\cdot)$ in the weight definition should be efficient. Formally, we state our conjecture.

CONJECTURE 3. *Consider the algorithm described in Section 3.1 with weight of node i at time t as*

$$W_i(t) \quad = \quad f(Q_i(\lfloor t \rfloor)). \tag{2}$$

Then, this algorithm is positive Harris recurrent as long as $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^o$ and $f(x) = \log\log(x+e)$.

This conjecture is empirically found to be true in the context of a specific class of network graph topologies (grid graph) as suggested in section 6. However, such a verification can only be accepted with partial faith.

4. TECHNICAL PRELIMINARIES

We present some known results about stationary distribution and convergence time (or mixing time) to stationary distribution for a specific class of finite-state Markov chains known as Glauber dynamics (or Metropolis-Hastings). As the reader will find, these results will play an important role in establishing the positive Harris recurrence of the network Markov chain.

4.1 Finite state Markov chain

Consider a time-homogeneous Markov chain over a finite state space Ω. Let the $|\Omega| \times |\Omega|$ matrix P be its transition probability matrix. If P is irreducible and aperiodic, then the Markov chain has an unique stationary distribution and it is ergodic in the sense that $\lim_{\tau\to\infty} P^\tau(j,i) \to \pi_i$ for any $i, j \in \Omega$. Here $\pi = [\pi_i]$ denotes the stationary distribution of the Markov chain. The adjoint of the transition matrix P, also called the time-reversal of P, is denoted by P^* and defined as: for any $i, j \in \Omega$, $\pi(i)P^*(i,j) = \pi(j)P(j,i)$. By definition, P^* has π as its stationary distribution. If $P = P^*$ then P is called *reversible*.

Our interest is in a specific irreducible, aperiodic and reversible Markov chain on the finite space $\Omega = \mathcal{I}(G)$, the set of independent sets of a given network graph $G = (V, E)$. This is also known as Glauber dynamics (or Metropolis-Hastings). We define it next.

DEFINITION 2 (GLAUBER DYNAMICS). *Consider a node weighted graph $G = (V, E)$ with $\boldsymbol{W} = [W_i]_{i\in V}$ the vector of node weights. Let $\mathcal{I}(G)$ denote the set of all independent sets of G. Then the Glauber dynamics on $\mathcal{I}(G)$ with weights given by $\boldsymbol{W}$, denoted by $GD(\boldsymbol{W})$, is the following Markov chain. Suppose the Markov chain is at state $\boldsymbol{\sigma} = [\sigma_i]_{i\in V}$, then the next transition happens as follows:*

- *Pick a node $i \in V$ uniformly at random.*
- *If $\sigma_j = 0$ for all $j \in \mathcal{N}(i)$, then*

$$\sigma_i = \begin{cases} 1 & \text{with probability } \frac{\exp(W_i)}{1+\exp(W_i)} \\ 0 & \text{otherwise.} \end{cases}$$

- *Otherwise, $\sigma_i = 0$.*

As the reader will notice, our algorithm described in Section 3 is effectively an asynchronous version of the above described Glauber dynamics with time-varying weights. In essence, we will be establishing that even with asynchronous time-varying weights, the behavior of our algorithm will be very close to that of the Glauber dynamics with fixed weight in its stationarity. To this end, next we state a property of this Glauber dynamics in terms of its stationary distribution, which follows easily from the reversibility of $GD(\boldsymbol{W})$.

LEMMA 4. *Let π be the stationary distribution of $GD(\boldsymbol{W})$ on the space of independent sets $\mathcal{I}(G)$ of the graph $G = (V, E)$. Then,*

$$\pi(\boldsymbol{\sigma}) = \frac{1}{Z} \exp(\boldsymbol{W} \cdot \boldsymbol{\sigma}) \cdot \mathbf{1}_{\boldsymbol{\sigma}\in\mathcal{I}(G)},$$

where Z is the normalizing factor.

4.2 Mixing time

The Glauber dynamics as described above converges to its stationary distribution π starting from any initial condition. To establish our results, we will need quantitative bounds on the time it takes for the Glauber dynamics to reach "close" to its stationary distribution. To this end, we start with the definition of distances between probability distributions.

DEFINITION 3. *(Distance of measures) Given two probability distributions ν and μ on a finite space Ω, we define the following two distances. The total variation distance, denoted as $\|\nu - \mu\|_{TV}$ is*

$$\|\nu - \mu\|_{TV} = \frac{1}{2} \sum_{i\in\Omega} |\nu(i) - \mu(i)| .$$

The χ^2 distance, *denoted as* $\left\|\frac{\nu}{\mu}-1\right\|_{2,\mu}$ *is*

$$\left\|\frac{\nu}{\mu}-1\right\|_{2,\mu}^2 = \|\nu-\mu\|_{2,\frac{1}{\mu}}^2 = \sum_{i\in\Omega}\mu(i)\left(\frac{\nu(i)}{\mu(i)}-1\right)^2.$$

More generally, for any two vectors $\mathbf{u},\mathbf{v}\in\mathbb{R}_+^{|\Omega|}$, *we define*

$$\|\mathbf{v}\|_{2,\mathbf{u}}^2 = \sum_{i\in\Omega} u_i v_i^2.$$

We make note of the following relation between the two distances defined above: using the Cauchy-Schwarz inequality, we have

$$\left\|\frac{\nu}{\mu}-1\right\|_{2,\mu} \geq 2\left\|\nu-\mu\right\|_{TV}. \tag{3}$$

Next, we define a matrix norm that will be useful in determining the rate of convergence or the mixing time of a finite-state Markov chain.

Definition 4 (Matrix norm). *Consider a* $|\Omega|\times|\Omega|$ *non-negative valued matrix* $A\in\mathbb{R}_+^{|\Omega|\times|\Omega|}$ *and a given vector* $\mathbf{u}\in\mathbb{R}_+^{|\Omega|}$. *Then, the matrix norm of* A *with respect to* $\mathbf{u}$ *is defined as follows:*

$$\|A\|_{\mathbf{u}} = \sup_{\mathbf{v}:\mathbb{E}_{\mathbf{u}}[\mathbf{v}]=0}\frac{\|A\mathbf{v}\|_{2,\mathbf{u}}}{\|\mathbf{v}\|_{2,\mathbf{u}}},$$

where $\mathbb{E}_{\mathbf{u}}[\mathbf{v}] = \sum_i u_i v_i$.

For a probability matrix P, we will mostly be interested in the matrix norm of P with respect to its stationary distribution π, i.e. $\|P\|_\pi$. Therefore, in this paper if we use a matrix norm for a probability matrix without mentioning the reference measure, then it is with respect to the stationary distribution. That is, in the above example $\|P\|$ will mean $\|P\|_\pi$.

With these definitions, it follows that for any distribution μ on Ω

$$\left\|\frac{\mu P}{\pi}-1\right\|_{2,\pi} \leq \|P^*\|\left\|\frac{\mu}{\pi}-1\right\|_{2,\pi}, \tag{4}$$

since $\mathbb{E}_\pi\left[\frac{\mu}{\pi}-1\right]=0$, where $\frac{\mu}{\pi}=[\mu(i)/\pi(i)]$. The Markov chain of our interest, Glauber dynamics, is reversible i.e. $P=P^*$. This suggests that in order to bound the distance between a Markov chain's distribution after some steps and its stationary distribution, it is sufficient to obtain a bound on $\|P\|$. One such bound can be obtained as below using Cheeger's inequality, and the details of its proof are omitted due to space constraints.

Lemma 5. *Let* P *be the transition matrix of the Glauber dynamics* $GD(\boldsymbol{W})$ *on a graph* $G=(V,E)$ *of* $n=|V|$ *nodes. Then,*

$$\|P\| \leq 1-\frac{1}{8n^2\exp(4n\overline{W}_{\max})},$$

where $\overline{W}_{\max}=\max\{1,W_{\max}\}$ *and* $W_{\max}=\max_{i\in V}W_i$.

5. PROOF OF MAIN RESULT

This section presents the detailed proof of Theorem 1. We will present the sketch of the proof followed by details.

5.1 Proof sketch

We first introduce the necessary definition of the network Markov process under our algorithm. As before, let $\tau\in\mathbb{N}$ be the index for discrete time. Let $\mathbf{Q}(\tau)=[Q_i(\tau)]$ denote the vector of queue-sizes at time τ, $\widetilde{\mathbf{Q}}(\tau)=[\widetilde{Q}_{\max,i}(\tau)]$ be the vector of estimates of $Q_{\max}(\tau)$ at time τ and $\boldsymbol{\sigma}(\tau)=[\sigma_i(\tau)]$ be the scheduling choices at the n nodes at time τ. Then it can be checked that the tuple $X(\tau)=(\mathbf{Q}(\tau),\widetilde{\mathbf{Q}}(\tau),\boldsymbol{\sigma}(\tau))$ is the Markov state of the network operating under the algorithm. Note that $X(\tau)\in\mathsf{X}$ where $\mathsf{X}=\mathbb{R}_+^n\times\mathbb{R}_+^n\times\mathcal{I}(G)$. Clearly, X is a Polish space endowed with the natural product topology. Let $\mathcal{B}_{\mathsf{X}}$ be the Borel σ-algebra of X with respect to this product topology. Let P denote the probability transition matrix of this discrete-time X-valued Markov chain. We wish to establish that $X(\tau)$ is indeed positive Harris recurrent under this setup. For any $\mathbf{x}=(\mathbf{Q},\widetilde{\mathbf{Q}},\boldsymbol{\sigma})\in\mathsf{X}$, we define norm of $\mathbf{x}$ denoted by $|\mathbf{x}|$ as

$$|\mathbf{x}| = |\mathbf{Q}|+|\widetilde{\mathbf{Q}}|+|\boldsymbol{\sigma}|,$$

where $|\mathbf{Q}|$ and $|\widetilde{\mathbf{Q}}|$ denote the standard ℓ_1 norm while $|\boldsymbol{\sigma}|$ is defined as its index in $\{0,\ldots,|\mathcal{I}(G)|-1\}$, which is assigned arbitrarily. Thus, $|\boldsymbol{\sigma}|$ is always bounded. Further, by Lemma 2, we have $|\widetilde{\mathbf{Q}}|\leq|\mathbf{Q}|$ under the evolution of Markov chain. Therefore, in essence $|\mathbf{x}|\to\infty$ if and only if $|\mathbf{Q}|\to\infty$. Next, we present the proof based on a sequence of lemmas. The proofs will be presented subsequently.

We will need some definitions to begin with. Given a probability distribution (also called sampling distribution) a on $\mathbb{N}$, the a-sampled transition matrix of the Markov chain, denoted by K_a is defined as

$$K_a(\mathbf{x},B)=\sum_{\tau\geq0}a(\tau)P^\tau(\mathbf{x},B),\quad\text{for any}\quad\mathbf{x}\in\mathsf{X},\ B\in\mathcal{B}_{\mathsf{X}}.$$

Now, we define a notion of a *petite* set. A non-empty set $A\in\mathcal{B}_{\mathsf{X}}$ is called μ_a-*petite* if μ_a is a non-trivial measure on $(\mathsf{X},\mathcal{B}_{\mathsf{X}})$ and a is a probability distribution on $\mathbb{N}$ such that for any $\mathbf{x}\in A$,

$$K_a(\mathbf{x},\cdot)\geq\mu_a(\cdot).$$

A set is called a *petite* set if it is μ_a-petite for some such non-trivial measure μ_a. A known sufficient condition to establish positive Harris recurrence of a Markov chain is to establish positive Harris recurrence of closed petite sets as stated in the following lemma. We refer an interested reader to the book by Meyn and Tweedie [23] or the recent survey by Foss and Konstantopoulos [8] for details.

Lemma 6. *Let* B *be a closed petite set. Suppose* B *is Harris recurrent, i.e.* $\Pr_{\mathbf{x}}(T_B<\infty)=1$ *for any* $\mathbf{x}\in\mathsf{X}$. *Further, let*

$$\sup_{\mathbf{x}\in B}\mathbb{E}_{\mathbf{x}}\left[T_B\right]<\infty.$$

Then the Markov chain is positive Harris recurrent.

Lemma 6 suggests that to establish the positive Harris recurrence of the network Markov chain, it is sufficient to find a closed petite set that satisfies the conditions of Lemma 6. To this end, we first establish that there exist closed sets that satisfy condition of Lemma 6. Later we will establish that they are indeed petite sets. This will conclude the proof of positive Harris recurrence of the network Markov chain.

Recall that the 'weight' function is $f(x) = \log\log(x+e)$. Define its integral, $F(x) = \int_0^x f(y)dy$. The system Lyapunov function, $L : \mathsf{X} \to \mathbb{R}_+$ is defined as

$$L(\mathbf{x}) = \sum_{i=1}^{n} F(Q_i) \triangleq F(\mathbf{Q}) \cdot \mathbf{1}, \quad \text{where } \mathbf{x} = (\mathbf{Q}, \widetilde{\mathbf{Q}}, \boldsymbol{\sigma}) \in \mathsf{X}.$$

We will establish the following, whose proof is given in Section 5.3.

LEMMA 7. *Let $\lambda \in (1-2\varepsilon)\Lambda^o$. Then there exist functions $h, g : \mathsf{X} \to \mathbb{R}$ such that for any $\mathbf{x} \in \mathsf{X}$,*

$$\mathbb{E}\left[L(X(g(\mathbf{x}))) - L(X(0)) | X(0) = \mathbf{x}\right] \le -h(\mathbf{x}),$$

and satisfy the following conditions: (a) $\inf_{\mathbf{x}\in\mathsf{X}} h(\mathbf{x}) > -\infty$, (b) $\liminf_{L(\mathbf{x})\to\infty} h(\mathbf{x}) > 0$, (c) $\sup_{L(\mathbf{x})\le\gamma} g(\mathbf{x}) < \infty$ for all $\gamma > 0$, and (d) $\limsup_{L(\mathbf{x})\to\infty} g(\mathbf{x})/h(\mathbf{x}) < \infty$.

Now define $B_\kappa = \{\mathbf{x} : L(\mathbf{x}) \le \kappa\}$ for any $\kappa > 0$. It will follow that B_κ is a closed set. Therefore, Lemma 7 and Theorem 1 in survey [8] imply that there exists constant $\kappa_0 > 0$ such that for all $\kappa_0 < \kappa$, the following holds:

$$\mathbb{E}_{\mathbf{x}}\left[T_{B_\kappa}\right] < \infty, \qquad \text{for any } \mathbf{x} \in \mathsf{X} \tag{5}$$

$$\sup_{\mathbf{x}\in B_\kappa} \mathbb{E}_{\mathbf{x}}\left[T_{B_\kappa}\right] < \infty. \tag{6}$$

Now we are ready to state the final nugget required in proving positive Harris recurrence as stated below.

LEMMA 8. *Consider any $\kappa > 0$. Then, the set $B_\kappa = \{\mathbf{x} : L(\mathbf{x}) \le \kappa\}$ is a closed petite set.*

The proof of Lemma 8 is technical and omitted due to space constraints. Lemmas 6, 7 and 8 imply that the network Markov chain is positive Harris recurrent. This completes the proof of Theorem 1.

5.2 Some preliminaries

Now we relate our algorithm described in Section 3.1 with an appropriate continuous time version of the Glauber dynamics described in Section 4.1. To this end, recall that the algorithm changes its scheduling decision when a node's Exponential clock of rate 1 ticks. Due to the property of the Exponential distribution, no two nodes have clocks ticking at the same time. Now given a clock tick, it is equally likely to be any of the n nodes. The node whose clock ticks, decides its transition based on probability prescribed by the Glauber dynamics $GD(\boldsymbol{W}(t))$ where recall that $\boldsymbol{W}(t)$ are determined based on $\mathbf{Q}(\lfloor t \rfloor), \widetilde{\mathbf{Q}}(\lfloor t \rfloor)$. Thus the transition probabilities of the Markov process determining the schedule $\boldsymbol{\sigma}(t)$ change every discrete time. Let $P(t)$ denote the transition matrix prescribed by the Glauber dynamics $GD(\boldsymbol{W}(t))$ and $\pi(t)$ denote its stationary distribution. Now the scheduling algorithm evolves the scheduling decision $\boldsymbol{\sigma}(\cdot)$ over time with time varying $P(t)$ as described before. Let $\mu(t)$ be the distribution of the schedule $\boldsymbol{\sigma}(t)$ at time t. The algorithm is essentially running $P(t)$ on $\mathcal{I}(G)$ when a clock ticks at time t. Since there are n clocks with rate 1 and $P(t) = P(\lfloor t \rfloor)$, we have

$$\begin{aligned}\mu(t) &= \sum_{i=0}^{\infty} \Pr(\zeta = i)\mu(\lfloor t \rfloor)P(\lfloor t \rfloor)^i \\ &= \mu(\lfloor t \rfloor)e^{n(t-\lfloor t \rfloor)(P(\lfloor t \rfloor)-I)}, \end{aligned} \tag{7}$$

where ζ be the number of clock ticks in time $(\lfloor t \rfloor, t]$ and it is distributed as a Poisson random variable with mean $n(t - \lfloor t \rfloor)$. Thus, for any $\tau \in \mathbb{N}$,

$$\mu(\tau+1) = \mu(\tau)e^{n(P(\tau)-I)}. \tag{8}$$

The equation (8) gives the discrete-time interpretation on μ, hence the mixing-time based analysis on μ with the transition matrix $e^{n(P(\tau)-I)}$ becomes possible. The transition matrix $e^{n(P(\tau)-I)}$ has properties similar to that of $P(\tau)$, as stated below. The details of its proof are omitted due to space constraints; they use Lemma 5 and properties of the matrix norm.

LEMMA 9. *$e^{n(P(\tau)-I)}$ is reversible and its stationary distribution is $\pi(\tau)$. Furthermore, its matrix norm is bounded as*

$$\left\| e^{n(P(\tau)-I)} \right\| \le 1 - \frac{1}{16n\exp(4n\overline{W}_{\max}(\tau))}.$$

5.3 Proof of Lemma 7

We have $\lambda \in (1-2\varepsilon)\Lambda^o$. That is, for some $\delta > 0$, $\boldsymbol{\lambda} \le (1-2\varepsilon-\delta)\Lambda$. The proof of Lemma 7 crucially utilizes the following Lemma 10, which we will prove in Section 5.4.

LEMMA 10. *For given $\delta, \varepsilon > 0$, let $\boldsymbol{\lambda} \le (1-2\varepsilon-\delta)\Lambda$. Define a large enough constant $B = B(n,\varepsilon)$ such that it satisfies the following:*

$$B \ge (16n-1)^{16n-1} \text{ and } \frac{256n^2\left(\log(x+e)\right)^{4n}}{e^{(\log(x-2n+e))^{\varepsilon/n}} - e - 1} < \varepsilon,$$

for all $x \ge B$. [3] Now, given any starting condition $\mathbf{x} = (\mathbf{Q}(0), \widetilde{\mathbf{Q}}(0), \boldsymbol{\sigma}(0))$, there exists a constant $C \triangleq C(\mathbf{Q}(0))$ such that for $T \in I \cap \mathbb{N}$ where $I = [C, Q_{\max}(0) - B]$,

$$\begin{aligned}&\mathbb{E}_{\mathbf{x}}[L(X(T)) - L(X(C))] \\ &\le -\frac{\delta}{n}\sum_{\tau=C}^{T-1} \mathbb{E}_{\mathbf{x}}[f(\boldsymbol{Q}(\tau)) \cdot \mathbf{1}] + 6n(T-C),\end{aligned}$$

with $C(\boldsymbol{Q}(0)) = O\left(\log^{16n+1} Q_{\max}(0)\right)$. Here, as usual, $\mathbb{E}_{\mathbf{x}}[\cdot]$ denotes expectation with respect to the condition that $X(0) = \mathbf{x}$.

Now proceed towards the proof of Lemma 7. We choose $g(\mathbf{x}) = \lceil \log Q_{\max}(0) + 2 \rceil C$. Since $g = O(C \log Q_{\max}(0)) = O\left(\log^{16n+2} Q_{\max}(0)\right)$, there exists a constant $D = D(n,\varepsilon,\delta)$ such that $g < Q_{\max}(0) - B$ whenever $Q_{\max}(0) \ge D$. Hence if $Q_{\max}(0) \ge D$, using Lemma 10,

$$\begin{aligned}&\mathbb{E}_{\mathbf{x}}[L(X(g(\mathbf{x}))) - L(X(0))] \\ &\le -\frac{\delta}{n}\sum_{\tau=C}^{g(\mathbf{x})-1} \mathbb{E}_{\mathbf{x}}[f(\boldsymbol{Q}(\tau)) \cdot \mathbf{1}] + 6n(g(\mathbf{x}) - C) \\ &\quad + \mathbb{E}_{\mathbf{x}}[L(X(C)) - L(X(0))] \\ &\le -\frac{\delta}{n}\sum_{\tau=C}^{g(\mathbf{x})-1} \mathbb{E}_{\mathbf{x}}[f(Q_{\max}(\tau))] + 6n(g(\mathbf{x}) - C) \\ &\quad + \mathbb{E}_{\mathbf{x}}[L(X(C)) - L(X(0))] \\ &\le -\frac{\delta}{n}(g(\mathbf{x}) - C) f\left((Q_{\max}(0) - g(\mathbf{x}))^+\right) + 6n(g(\mathbf{x}) - C) \\ &\quad + \mathbb{E}_{\mathbf{x}}[F(\boldsymbol{Q}(C)) \cdot \mathbf{1} - F(\boldsymbol{Q}(0)) \cdot \mathbf{1}] \\ &\le -\frac{\delta}{n}(g(\mathbf{x}) - C) f\left((Q_{\max}(0) - g(\mathbf{x}))^+\right) + 6n(g(\mathbf{x}) - C) \\ &\quad + C\, n\, f(Q_{\max}(0) + C) \\ &\triangleq k(\mathbf{x}).\end{aligned} \tag{9}$$

[3] There exists such a constant B since $\lim_{x\to\infty} \frac{256n^2(\log(x+e))^{4n}}{e^{(\log(x-2n+e))^{\varepsilon/n}} - e - 1} = 0$ for any fixed $n, \varepsilon > 0$.

When $Q_{\max}(0) \le D$, $\mathbb{E}_{\mathbf{x}}[L(X(g(\mathbf{x}))) - L(X(0))]$ is bounded by a constant $E = E(n, \varepsilon, \delta)$ since $g(\mathbf{x})$ is bounded in terms of $Q_{\max}(0)$ and $Q_{\max}(g(\mathbf{x})) \le Q_{\max}(0) + g(\mathbf{x})$. Therefore, we can define functions h as follows

$$h(\mathbf{x}) = \begin{cases} -k(\mathbf{x}) & \text{if } Q_{\max}(0) \ge D \\ -E & \text{otherwise} \end{cases},$$

which satisfies

$$\mathbb{E}\left[L(X(g(\mathbf{x}))) - L(X(0)) | X(0) = \mathbf{x}\right] \quad \le \quad -h(\mathbf{x}).$$

The desired conditions of Lemma 7 can be checked as: (c) is trivial and (a), (b) and (d) follow since h/g grows in order of $f(Q_{max}(0))$ for a large $Q_{max}(0)$ due to our choice of $g(\mathbf{x})$.

5.4 Proof of Lemma 10

Here we prove Lemma 10 using the following two lemmas.

LEMMA 11. *Consider a vector of queue-sizes $\mathbf{Q} \in \mathbb{R}_+^n$. Let the vector of estimation of $Q_{\max}$ be $\widetilde{\mathbf{Q}} \in \mathbb{R}_+^n$ satisfying the property of Lemma 2. Let weight vector $\boldsymbol{W}$ based on these queues be defined as per equation (1). Consider the Glauber dynamics $GD(\boldsymbol{W})$ and let π denote its stationary distribution. If $\boldsymbol{\sigma}$ is distributed as per π then*

$$\mathbb{E}_{\pi}[f(\boldsymbol{Q}) \cdot \boldsymbol{\sigma}] \ge (1-\varepsilon)\left(\max_{\boldsymbol{\rho} \in \mathcal{I}(G)} f(\mathbf{Q}) \cdot \boldsymbol{\rho}\right) - 3n.$$

The proof of Lemma 11 is omitted due to space constraints.

LEMMA 12 (NETWORK ADIABATIC THEOREM). *For any given $\mathcal{F} = \{\boldsymbol{Q}(\tau), \widetilde{\mathbf{Q}}(\tau) | \tau = 0, 1, \ldots \lfloor t \rfloor\}$, let $\bar{\mu}(t)$ be the (conditional) distribution of the schedule over $\mathcal{I}(G)$ at time t, and let $\pi(t)$ be the stationary distribution of the Markov process over $\mathcal{I}(G)$ given by the probability transition matrix $P(t)$ as defined in Section 5.2. Then, for $t \in I = [C_1(Q_{\max}(0)), Q_{\max}(0) - B]$,*

$$\left\| \frac{\bar{\mu}(t)}{\pi(t)} - 1 \right\|_{2,\pi(t)} < \varepsilon, \qquad \textit{with probability } 1,$$

where $C_1(x)$ is given by

$$\left\lceil 16^2 n^2 \log^{8n}(x+1+e) \log\left(\frac{2}{\varepsilon}\left(2\log(x+e)\right)^{n/2}\right)\right\rceil^2 + 1.$$

REMARK 1. *$\bar{\mu}(t)$ and $\pi(t)$ are random variables depending on $\mathcal{F}$, hence $\mu(t) = \mathbb{E}[\bar{\mu}(t)]$ where the expectation is taken over the distribution of $\mathcal{F}$. The statement of Lemma 12 suggests that, if queue-sizes are large, the distribution $\bar{\mu}(t)$ of schedules is essentially close to the stationary distribution $\pi(t)$ for large enough time, despite the fact that the weights (or queue-sizes) keep changing.*

The proof of Lemma 12 is presented in Section 5.5. Now we proceed towards proving Lemma 10. From Lemma 12 and relation (3) we have that for $t \in I$,

$$\left|\mathbb{E}_{\pi(t)}[f(\boldsymbol{Q}(t)) \cdot \boldsymbol{\sigma}] - \mathbb{E}_{\bar{\mu}(t)}[f(\boldsymbol{Q}(t)) \cdot \boldsymbol{\sigma}]\right| \le \varepsilon \left(\max_{\boldsymbol{\rho} \in \mathcal{I}(G)} f(\boldsymbol{Q}(t)) \cdot \boldsymbol{\rho}\right).$$

Thus from Lemma 11,

$$\mathbb{E}_{\bar{\mu}(t)}[f(\boldsymbol{Q}(t)) \cdot \boldsymbol{\sigma}] \ge (1-2\varepsilon)\left(\max_{\boldsymbol{\rho} \in \mathcal{I}(G)} f(\boldsymbol{Q}(t)) \cdot \boldsymbol{\rho}\right) - 3n.$$

Now we can bound the difference between $L(X(\tau+1))$ and $L(X(\tau))$ as follows.

$$\begin{aligned}
&L(X(\tau+1)) - L(X(\tau)) \\
&= (F(\boldsymbol{Q}(\tau+1)) - F(\boldsymbol{Q}(\tau))) \cdot \mathbf{1} \\
&\le f(\boldsymbol{Q}(\tau+1)) \cdot (\boldsymbol{Q}(\tau+1) - \boldsymbol{Q}(\tau)), \qquad \text{(as } F \text{ is convex)}, \\
&\overset{(a)}{\le} f(\boldsymbol{Q}(\tau)) \cdot (\boldsymbol{Q}(\tau+1) - \boldsymbol{Q}(\tau)) + n \\
&= f(\boldsymbol{Q}(\tau)) \cdot \left(A(\tau, \tau+1) - \int_{\tau}^{\tau+1} \boldsymbol{\sigma}(y) \mathbf{1}_{\{Q_i(y) > 0\}}\, dy\right) + n \\
&\overset{(b)}{\le} f(\boldsymbol{Q}(\tau)) \cdot A(\tau, \tau+1) - \int_{\tau}^{\tau+1} f(\boldsymbol{Q}(y)) \cdot \boldsymbol{\sigma}(y) \mathbf{1}_{\{Q_i(y) > 0\}}\, dy + 2n \\
&= f(\boldsymbol{Q}(\tau)) \cdot A(\tau, \tau+1) - \int_{\tau}^{\tau+1} f(\boldsymbol{Q}(y)) \cdot \boldsymbol{\sigma}(y)\, dy + 2n, \qquad (10)
\end{aligned}$$

where (a) and (b) follow from the fact that f is 1-Lipschitz[4] and $\mathbf{Q}(\cdot)$ changes at unit rate. For $\tau, \tau+1 \in I$, if we take the expectation of (10) over the distribution of $\boldsymbol{\sigma}$ given $\mathcal{F}$, we have

$$\begin{aligned}
&\mathbb{E}[L(X(\tau+1)) - L(X(\tau)) \,|\, \mathcal{F}] \\
&\le \mathbb{E}[f(\boldsymbol{Q}(\tau)) \cdot A(\tau, \tau+1) \,|\, \mathcal{F}] - \int_{\tau}^{\tau+1} \mathbb{E}_{\bar{\mu}(t)}[f(\boldsymbol{Q}(y)) \cdot \boldsymbol{\sigma}(y)]\, dy + 2n \\
&\le f(\boldsymbol{Q}(\tau)) \cdot \boldsymbol{\lambda} - \int_{\tau}^{\tau+1} (1-2\varepsilon) \mathbb{E}\left[\max_{\boldsymbol{\rho} \in \mathcal{I}(G)} f(\boldsymbol{Q}(y)) \cdot \boldsymbol{\rho}\right] dy + 5n \\
&\le f(\boldsymbol{Q}(\tau)) \cdot \boldsymbol{\lambda} - (1-2\varepsilon)\left(\max_{\boldsymbol{\rho} \in \mathcal{I}(G)} f(\boldsymbol{Q}(\tau)) \cdot \boldsymbol{\rho}\right) + 6n \\
&\le -\delta \left(\max_{\boldsymbol{\rho} \in \mathcal{I}(G)} f(\boldsymbol{Q}(\tau)) \cdot \boldsymbol{\rho}\right) + 6n,
\end{aligned}$$

where the last inequality follows from $\boldsymbol{\lambda} \in (1-2\varepsilon-\delta)\boldsymbol{\Lambda}$. If we take the expectation again over the distribution of $\mathcal{F}$ given the initial state $X(0) = \mathbf{x}$, we obtain

$$\begin{aligned}
\mathbb{E}_{\mathbf{x}}[L(X(\tau+1)) - L(X(\tau))] &\le -\delta \mathbb{E}_{\mathbf{x}}\left[\max_{\boldsymbol{\rho} \in \mathcal{I}(G)} f(\boldsymbol{Q}(\tau)) \cdot \boldsymbol{\rho}\right] dy + 6n \\
&\overset{(a)}{\le} -\frac{\delta}{n} \mathbb{E}_{\mathbf{x}}\left[f(\boldsymbol{Q}(\tau)) \cdot \mathbf{1}\right] dy + 6n.
\end{aligned}$$

In above, for (a), we use the fact that $\mathbf{1}$ can be written as a convex combination of n singleton independent sets. Therefore, by summing over τ from $C_1 = C_1(Q_{\max}(0))$ to $T-1$, we have

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}}\left[L(X(T)) - L(X(C_1))\right] \\
&\le -\frac{\delta}{n} \sum_{\tau=C_1}^{T-1} \mathbb{E}_{\mathbf{x}}[f(\boldsymbol{Q}(\tau)) \cdot \mathbf{1}] + 6n(T - C_1).
\end{aligned}$$

From Lemma 12, by the choice of $C(\mathbf{Q}(0)) = C_1(Q_{\max}(0)) = O\left(\log^{16n+1} Q_{\max}(0)\right)$, we obtain the desired result and complete the proof of Lemma 10.

5.5 Proof of Network adiabatic theorem

This section establishes the proof of Lemma 12. In words, Lemma 12 states that, if queue-sizes are large, the observed distribution of schedules is essentially the same as the desired stationary distribution for large enough time despite the fact that the weights (or queue-sizes) keep changing. In a nutshell, by selecting the weight function $f(\cdot) = \log\log(\cdot + e)$, the dynamics of weights become "slow enough", thus allowing for the distribution of scheduling decisions to remain

[4] A continuous function $f : \mathbb{R} \to \mathbb{R}$ is K-Lipschitz if $|f(x) - f(y)| \le K|x-y|$ for all $x, y \in \mathbb{R}$.

close to the desired stationary distribution. This is analogous to the classical adiabatic theorem which states that *if the system is changed gradually (slowly) in a reversible manner and if the system starts in the ground states then it remains in the ground state.*

5.5.1 Two useful results

We state two lemmas that will be useful for establishing Lemma 12. Before we state these lemmas, we define a transformation of the queue-size vector: define $\widehat{Q}_i = f^{-1}(W_i)$ and let $\widehat{\mathbf{Q}}$ be its corresponding vector.

LEMMA 13. *Given $\tau \in \mathbb{N}$, define*

$$\alpha_\tau = \left(f'(\widehat{\boldsymbol{Q}}(\tau)) + f'(\widehat{\boldsymbol{Q}}(\tau+1))\right)\cdot \mathbf{1}.$$

Then if $\alpha_\tau < 1$, the following hold:

1. *For any $\boldsymbol{\rho} \in \mathcal{I}(G)$, $\exp(-\alpha_\tau) \le \frac{\pi(\tau+1)(\boldsymbol{\rho})}{\pi(\tau)(\boldsymbol{\rho})} \le \exp(\alpha_\tau)$.*
2. *And, $\|\pi(\tau+1)-\pi(\tau)\|_{2,\frac{1}{\pi(\tau+1)}} \le 2\alpha_\tau$.*

The proof of Lemma 13 is quite standard and omitted due to space constraints. Next, we state a lemma which implies that the change in $\pi(\cdot)$ is "small" compared to the "mixing time" of the Glauber dynamics when the queue-size is large. It will play a crucial role in establishing Lemma 12, and the appropriate choice of large enough B in Lemma 10 is necessary for its proof.

LEMMA 14. *If $Q_{\max}(\tau+1) \ge B$,*

$$T_{\tau+1}\alpha_\tau \le \frac{\varepsilon}{8}, \tag{11}$$

where T_τ stands for the mixing time of the transition matrix $e^{n(P(\tau)-I)}$ and is defined as $T_\tau = \frac{1}{1-\|e^{n(P(\tau)-I)}\|}$. Note that $\alpha_\tau < T_{\tau+1}\alpha_\tau \le \frac{\varepsilon}{8} < \frac{\varepsilon}{4+\varepsilon} < 1$.

PROOF. First note that

$$\begin{aligned}\widehat{Q}_{\min} &= f^{-1}(W_{\min}) \ge f^{-1}\left(\frac{\varepsilon}{n} f(\widetilde{Q}_{\max,i})\right)\\ &\ge f^{-1}\left(\frac{\varepsilon}{n} f\left((Q_{\max}-2n)^+\right)\right)\\ &= f^{-1}\left(\frac{\varepsilon}{n} f\left((\widehat{Q}_{\max}-2n)^+\right)\right),\end{aligned}$$

from $Q_{\max} = \widehat{Q}_{\max}$ and Lemma 2. From Lemma 9, we have that $T_{\tau+1} \le 16n\left(\log\left(\widehat{Q}_{\max}(\tau+1)+e\right)\right)^{4n}$. Additionally by using $f'(x) = \frac{1}{(x+e)\log(x+e)} < \frac{1}{x}$, the following bound can be obtained:

$$\begin{aligned}&T_{\tau+1}\alpha_\tau\\ &\le 16n\log^{4n}\left(\widehat{Q}_{\max}(\tau+1)+e\right)\left[\left(f'(\widehat{\boldsymbol{Q}}(\tau)) + f'(\widehat{\boldsymbol{Q}}(\tau+1))\right)\cdot\mathbf{1}\right]\\ &\le 16n\log^{4n}\left(\widehat{Q}_{\max}(\tau+1)+e\right)\left(\frac{n}{\widehat{Q}_{\min}(\tau)} + \frac{n}{\widehat{Q}_{\min}(\tau+1)}\right)\\ &\le \frac{32n^2\log^{4n}\left(\widehat{Q}_{\max}(\tau+1)+e\right)}{\widehat{Q}_{\min}(\tau+1)-1} \qquad (\text{as } \widehat{Q} \text{ is 1-Lipschitz})\\ &\le \frac{32n^2\log^{4n}\left(\widehat{Q}_{\max}(\tau+1)+e\right)}{f^{-1}\left(\frac{\varepsilon}{n}f\left(\widehat{Q}_{\max}(\tau+1)-2n\right)\right)-1} \qquad (\text{from (12)})\\ &\le \frac{32n^2\log^{4n}(x+e)}{e^{(\log(x-2n+e))^{\frac{\varepsilon}{n}}}-e-1}, \end{aligned}\tag{12}$$

where $x := \widehat{Q}_{\max}(\tau+1) \ge B$. By our choice of B in Lemma 10, the right hand side of (12) is bounded above by $\varepsilon/8$. This completes the proof of Lemma 14. □

5.5.2 Proof of Lemma 12

For simplifying notations, let $\mu(t) = \bar{\mu}(t)$ in this section. First note that we can assume $0 < C_1(Q_{\max}(0)) \le Q_{\max}(0) - B$. Otherwise the conclusion is trivial since I is empty. We wish to establish that for $t \in I$,

$$\left\|\frac{\mu(t)}{\pi(t)} - 1\right\|_{2,\pi(t)} < \varepsilon.$$

It is enough to show the statement for $\tau = \lfloor t \rfloor \in I$ since

$$\begin{aligned}&\left\|\frac{\mu(t)}{\pi(t)} - 1\right\|_{2,\pi(t)} = \left\|\frac{\mu(t)}{\pi(\lfloor t\rfloor)} - 1\right\|_{2,\pi(\lfloor t\rfloor)} \qquad (\text{as } \pi(t) = \pi(\lfloor t\rfloor))\\ &\le \left\|e^{n(t-\lfloor t\rfloor)(P(\lfloor t\rfloor)-I)}\right\| \left\|\frac{\mu(\lfloor t\rfloor)}{\pi(\lfloor t\rfloor)} - 1\right\|_{2,\pi(\lfloor t\rfloor)} \qquad (\text{from (7)})\\ &\le \left\|\frac{\mu(\lfloor t\rfloor)}{\pi(\lfloor t\rfloor)} - 1\right\|_{2,\pi(\lfloor t\rfloor)} < \varepsilon.\end{aligned}$$

Now we first show that for any $\tau \in \mathbb{N}$ with $\tau+1 \in I$,

$$\left\|\frac{\mu(\tau+1)}{\pi(\tau)} - 1\right\|_{2,\pi(\tau)} < \varepsilon/2. \tag{13}$$

Suppose (13) is correct. Then, for $\tau \in I$, using (13), Lemmas 13 and 14, one can obtain the desired bound:

$$\begin{aligned}&\left\|\frac{\mu(\tau)}{\pi(\tau)} - 1\right\|_{2,\pi(\tau)} = \|\mu(\tau)-\pi(\tau)\|_{2,\frac{1}{\pi(\tau)}}\\ &\le (e^{\alpha_{\tau-1}/2})\,\|\mu(\tau)-\pi(\tau)\|_{2,\frac{1}{\pi(\tau-1)}}\\ &\overset{(a)}{\le} (1+\alpha_{\tau-1})\,\|\mu(\tau)-\pi(\tau)\|_{2,\frac{1}{\pi(\tau-1)}}\\ &\le (1+\alpha_{\tau-1})\Big(\|\mu(\tau)-\pi(\tau-1)\|_{2,\frac{1}{\pi(\tau-1)}}\\ &\qquad + \|\pi(\tau-1)-\pi(\tau)\|_{2,\frac{1}{\pi(\tau-1)}}\Big)\\ &\le (1+\alpha_{\tau-1})\left(\frac{\varepsilon}{2}+2\alpha_{\tau-1}\right) \overset{(b)}{\le} \left(1+\frac{\varepsilon}{8}\right)\left(\frac{\varepsilon}{2}+\frac{\varepsilon}{4}\right) < \varepsilon,\end{aligned}\tag{14}$$

where (a) and (b) are due to $\alpha_{\tau-1} \le \varepsilon/8 < 1$ from $Q_{\max}(\tau) \ge Q_{\max}(0) - \tau \ge B$ and Lemma 14. Therefore, it suffices to establish (13) for completing the proof of Lemma 12. For simplicity of notation, define

$$a_\tau \triangleq \left\|\frac{\mu(\tau+1)}{\pi(\tau)} - 1\right\|_{2,\pi(\tau)}.$$

Consider the following recursive relation for a_τ:

$$\begin{aligned}a_{\tau+1} &= \left\|\frac{\mu(\tau+2)}{\pi(\tau+1)} - 1\right\|_{2,\pi(\tau+1)}\\ &\le \left\|e^{n(P(\tau+1)-I)}\right\| \left\|\frac{\mu(\tau+1)}{\pi(\tau+1)} - 1\right\|_{2,\pi(\tau+1)} \qquad (\text{from (8)})\\ &= \left(1-\frac{1}{T_{\tau+1}}\right)\|\mu(\tau+1)-\pi(\tau+1)\|_{2,\frac{1}{\pi(\tau+1)}}\\ &\le \left(1-\frac{1}{T_{\tau+1}}\right)\Big(\|\mu(\tau+1)-\pi(\tau)\|_{2,\frac{1}{\pi(\tau+1)}}\\ &\qquad + \|\pi(\tau)-\pi(\tau+1)\|_{2,\frac{1}{\pi(\tau+1)}}\Big)\\ &\le \left(1-\frac{1}{T_{\tau+1}}\right)\left(e^{\alpha_\tau/2}\,\|\mu(\tau+1)-\pi(\tau)\|_{2,\frac{1}{\pi(\tau)}} + 2\alpha_\tau\right)\\ &\le \left(1-\frac{1}{T_{\tau+1}}\right)\left((1+\alpha_\tau)\,a_\tau + 2\alpha_\tau\right),\end{aligned}\tag{15}$$

where each inequality can be derived similarly as we derived (14). From (15), if we have $a_\tau < \varepsilon/2$ and $\tau+1 \in I$, then

$$\begin{aligned} a_{\tau+1} &< \left(1-\frac{1}{T_{\tau+1}}\right)(\varepsilon/2+(2+\varepsilon/2)\alpha_\tau) \\ &\leq \left(1-\frac{1}{T_{\tau+1}}\right)\left(\varepsilon/2+\frac{\varepsilon}{2T_{\tau+1}}\right) \quad \text{(from Lemma 14)} \\ &< \varepsilon/2. \end{aligned} \tag{16}$$

Hence, for establishing (13) it is enough to show that there exists a C such that $a_C < \varepsilon/2$ and $C \leq C_1(Q_{\max}(0))-1$. To this end, fix $\tau < Q_{\max}(0)-B$ and assume $a_s \geq \varepsilon/2$ for all integers $s < \tau$. Then, from (15),

$$\begin{aligned} a_\tau &\leq \left(1-\frac{1}{T_\tau}\right)((1+\alpha_{\tau-1})\,a_{\tau-1}+2\alpha_{\tau-1}) \\ &\leq \left(1-\frac{1}{T_\tau}\right)\left((1+\alpha_{\tau-1})\,a_{\tau-1}+4\alpha_{\tau-1}\frac{a_{\tau-1}}{\varepsilon}\right) \\ &= \left(1-\frac{1}{T_\tau}\right)\left(1+\left(1+\frac{4}{\varepsilon}\right)\alpha_{\tau-1}\right)a_{\tau-1} \\ &\leq \left(1-\frac{1}{T_\tau}\right)\left(1+\frac{1}{T_\tau}\right)a_{\tau-1} \quad \text{(from Lemma 14)} \\ &< e^{-\frac{1}{T_\tau^2}}\,a_{\tau-1} \\ &< e^{-\sum_{s=1}^{\tau}\frac{1}{T_s^2}}\,a_0. \end{aligned} \tag{17}$$

Now, $\sum_{s=1}^{\tau}\frac{1}{T_s^2}$ can be bounded as:

$$\begin{aligned} \sum_{s=1}^{\tau}\frac{1}{T_s^2} &\geq \sum_{s=1}^{\tau}\frac{1}{16^2n^2e^{8nf(Q_{\max}(s))}} \quad \text{(from Lemma 9)} \\ &= \frac{1}{16^2n^2}\sum_{s=1}^{\tau}\left(\frac{1}{\log\left(Q_{\max}(s)+e\right)}\right)^{8n} \\ &\geq \frac{1}{16^2n^2}\sum_{s=1}^{\tau}\left(\frac{1}{\log\left(Q_{\max}(0)+s+e\right)}\right)^{8n} \\ &> \frac{\tau}{16^2n^2\left(\log\left(Q_{\max}(0)+\tau+e\right)\right)^{8n}} \\ &\overset{(a)}{\geq} \frac{\sqrt{\tau}}{16^2n^2\left(\log\left(Q_{\max}(0)+1+e\right)\right)^{8n}}, \end{aligned}$$

where (a) follows from the fact that $\tau \geq 1, Q_{\max}(0) \geq B \geq (16n-1)^{16n-1}$ and

$$\sqrt{x} \geq \left(\frac{\log(x+y)}{\log(1+y)}\right)^{8n}, \quad \forall x \geq 1, y \geq (16n-1)^{16n-1}.$$

Finally, a_0 is also bounded above as:

$$\begin{aligned} a_0 &= \left\|\frac{\mu(1)}{\pi(0)}-1\right\|_{2,\pi(0)} < \sqrt{\frac{1}{\pi_{\min}(0)}} < \sqrt{Z(0)} \\ &\leq \sqrt{2^n e^{nf(Q_{\max}(0))}} \leq \left(2\log(Q_{\max}(0)+e)\right)^{n/2}. \end{aligned}$$

Now if we choose C as

$$\left\lceil 16^2n^2\log^{8n}(Q_{\max}(0)+1+e)\log\left(\frac{2}{\varepsilon}\left(2\log(Q_{\max}(0)+e)\right)^{n/2}\right)\right\rceil^2,$$

it can be checked that $e^{-\sum_{i=1}^{C}\frac{1}{T_i^2}}a_0 < \varepsilon/2$. So from (17), if $a_s \geq \varepsilon/2$ for all $s < C$, $a_C < e^{-\sum_{i=1}^{C}\frac{1}{T_i^2}}a_0 < \varepsilon/2$. Otherwise, there exists $C' < C$ such that $a_{C'} < \varepsilon/2$, which also implies $a_C < \varepsilon/2$ from (16). In either case, $a_C < \varepsilon/2$ and it completes the proof of (13) and hence the proof of Lemma 12.

6. SIMULATION RESULTS

Setup. We consider a $N \times N$ two-dimensional grid graph topology to understand the performance of our algorithm. The selection of such a topology is for two reasons: One, due to the *bipartite* nature of the grid graph, we have a precise characterization of the capacity region, denoted by $\Lambda(1)$, given by

$$\Lambda(\rho) = \{\lambda : \lambda_u + \lambda_v \leq \rho \text{ for all edges } (u,v)\}$$

where $\rho \in [0,1]$ is the load and λ_v is the arrival rate at node v. Two, it is a reasonable approximation of the wireless network arising in the *mesh network* scenario. We assume arrival process to be Bernoulli. For a given load $\rho \in [0,1]$, we consider two traffic patterns: (1) *Uniform traffic*, where $\lambda_u = \rho/2$ for all u; and (2) *chessboard traffic*, where for $u=(i,j)$, $\lambda_u = 2\rho/3$ if $i+j$ is even, and $\rho/3$ otherwise.

Results/observations. Our algorithm is, in essence, a *learning* mechanism that tries to find *good* schedules. For that reason, the uniform traffic pattern is good as there are many *options* for good independent sets and hence it is easier for an algorithm to learn them. On the other hand, the chessboard pattern is much harder as it requires the algorithm to essentially select one of the few good schedules almost all the time. Indeed, our simulation results verify this intuition. For uniform traffic, the algorithm does very well. Due to space constraints, we therefore present results for the chessboard traffic only.

Here we report results for $N = 10$ (total $N^2 = 100$ nodes) and different loading $\rho = 0.5, 0.6, 0.7, 0.8$ for algorithm that uses weight $f(x) = \log\log(x+e)$ along with adjustment using $\widetilde{Q}_{\max,i}(\cdot)$.The time-evolution of the total queue-size over the whole network is presented in Figure 1. As the reader will notice, the algorithm keeps queue-sizes stable as expected for all loads. We observe that the algorithm without using information of estimation of $\widetilde{Q}_{\max}(\cdot)$ has essentially identical performance (see performance of log log weight in Figure 2)! And thus it supports our conjecture.

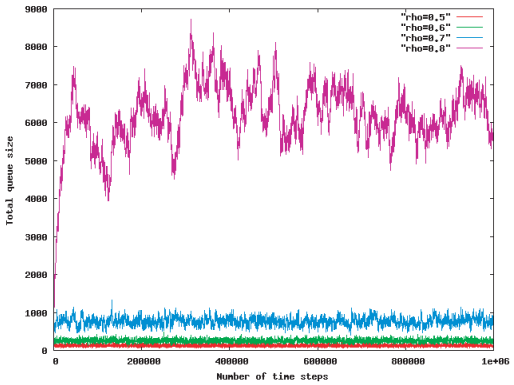

Figure 1: The evolution of queue-size with time, for different arrival rates

Finally, we try to understand the effect of the weight function f: we simulate for $f(x) = x, \log(x+1)$ and $\log\log(x+e)$. As expected, we find that for $f(x) = x$, system is clearly unstable (we do not report here due to space constraints). A comparison of log and log log (without any $\widetilde{Q}_{\max,i}(\cdot)$ based modification) weight functions is presented in Figure 2. It

clearly shows that the algorithm is stable for both of these weight functions; it is more *stable* (milder oscillations) for $\log\log$ compared to log weight but at the cost of higher queue-sizes. This plot clearly explains the effect of the selection of weight function: for stability, slowly changing weight function is necessary (i.e. $\log x$ or $\log\log x$ but not x); and among such functions slower function (i.e. $\log\log$ compared to log) leads to more stable network but at the cost of increased queue-sizes.

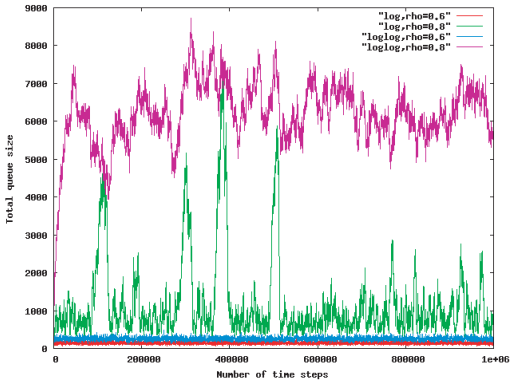

Figure 2: A comparison of log **and** log log **policies**

7. CONCLUSION

In this paper, we resolved the long-standing and important question of designing an efficient random-access algorithm for contention resolution in a network of queues. Our algorithm is essentially a random-access based implementation, inspired by Metropolis-Hastings sampling method, of the classical maximum weight algorithm with "weight" being an appropriate function ($f(x) = \log\log(x+e)$) of the queue-size. The key ingredient in establishing the efficiency of the algorithm is a novel *adiabatic*-like theorem for the underlying queueing network. We strongly believe that this *network adiabatic theorem* in particular and methods of this paper in general will be of interest in understanding the effect of dynamics in networked system.

8. REFERENCES

[1] N. Abramson and F. Kuo (Editors). The aloha system. *Computer-Communication Networks*, 1973.

[2] D. J. Aldous. Ultimate instability of exponential back-off protocol for acknowledgement-based transmission control of random access communication channels. *IEEE Transactions on Information Theory*, 33(2):219–223, 1987.

[3] C. Bordenave, D. McDonald, and A. Proutiere. Performance of random medium access - an asymptotic approach. In *Proceedings of ACM Sigmetrics*, 2008.

[4] M. Born and V. A. Fock. Beweis des adiabatensatzes. *Zeitschrift für Physik a Hadrons and Nuclei*, 51(3-4):165Ű180, 1928.

[5] J. G. Dai. Stability of fluid and stochastic processing networks. *Miscellanea Publication*, (9), 1999.

[6] A. Ephremides and B. Hajek. Information theory and communication networks: an unconsummated union. *IEEE Transactions on Information Theory*, 44(6):2416–2432, 1998.

[7] A. Eryilmaz, A. Ozdaglar, D. Shah, and E. Modiano. Distributed cross-layer algorithms for the optimal control of multi-hop wireless networks. *submitted to IEEE/ACM Transactions on Networking*, 2008.

[8] S. Foss and Takis Konstantopoulos. An overview of some stochastic stability methods. *Journal of Operations Research, Society of Japan*, 47(4), 2004.

[9] R. K. Getoor. Transience and recurrence of markov processes. *In AzŐma, J. and Yor, M., editors, Séminaire de Probabilités XIV*, pages 397–409, 1979.

[10] L.A. Goldberg, M. Jerrum, S. Kannan, and M. Paterson. A bound on the capacity of backoff and acknowledgement-based protocols. *Research Report 365, Department of Computer Science, University of Warwick, Coventry CV4 7AL, UK*, January 2000.

[11] Leslie Ann Goldberg. Design and analysis of contention-resolution protocols, epsrc research grant gr/l60982. http://www.csc.liv.ac.uk/ leslie/contention.html, Last updated, Oct. 2002.

[12] A. G. Greenberg, P. Flajolet, and R. E. Ladner. Estimating the multiplicities of conflicts to speed their resolution in multiple access channels. *Journal of the ACM*, 34(2):289–325, 1987.

[13] D. J. Griffiths. *Introduction to Quantum Mechanics*. Pearson Prentice Hall, 2005.

[14] P. Gupta and A. L. Stolyar. Optimal throughput allocation in general random-access networks. In *Proceedings of 40th Annual Conf. Inf. Sci. Systems, IEEE, Princeton, NJ*, pages 1254–1259, 2006.

[15] Johan Hastad, Tom Leighton, and Brian Rogoff. Analysis of backoff protocols for multiple access channels. *SIAM J. Comput*, 25(4), 1996.

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

[17] J.Liu and A. L. Stolyar. Distributed queue length based algorithms for optimal end-to-end throughput allocation and stability in multi-hop random access networks. In *Proceedings of 45th Allerton Conference on Communication, Control, and Computing, Urbana-Champaign, IL*, 2007.

[18] F. P. Kelly. Stochastic models of computer communication systems. *J. R. Statist. Soc B*, 47(3):379–395, 1985.

[19] F.P. Kelly and I.M. MacPhee. The number of packets transmitted by collision detect random access schemes. *The Annals of Probability*, 15(4):1557–1568, 1987.

[20] I.M. MacPhee. On optimal strategies in stochastic decision processes, d. phil. thesis, university of cambridge, 1989.

[21] P. Marbach. Distributed scheduling and active queue management in wireless networks. In *Proceedings of IEEE INFOCOM, Minisymposium*, 2007.

[22] P. Marbach, A. Eryilmaz, and A. Ozdaglar. Achievable rate region of csma schedulers in wireless networks with primary interference constraints. In *Proceedings of IEEE Conference on Decision and Control*, 2007.

[23] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Springer-Verlag, London, 1993.

[24] E. Modiano, D. Shah, and G. Zussman. Maximizing throughput in wireless network via gossiping. In *ACM SIGMETRICS/Performance*, 2006.

[25] Microsoft research lab. self organizing neighborhood wireless mesh networks. http://research.microsoft.com/mesh/.

[26] S. Sanghavi, L. Bui, and R. Srikant. Distributed link scheduling with constant overhead. In *Proceedings of ACM Sigmetrics*, 2007.

[27] D. Shah and D. J. Wischik. Optimal scheduling algorithm for input queued switch. In *Proceeding of IEEE INFOCOM*, 2006.

[28] D. Shah and D. J. Wischik. Heavy traffic analysis of optimal scheduling algorithms for switched networks. *Submitted*, 2007.

[29] S. Shakkottai and R. Srikant. *Network Optimization and Control*. Foundations and Trends in Networking, NoW Publishers, 2007.

[30] A. L. Stolyar. Dynamic distributed scheduling in random access networks. *Journal of Applied Probabability*, 45(2):297–313, 2008.

[31] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37:1936–1948, 1992.

[32] B.S. Tsybakov and N. B. Likhanov. Upper bound on the capacity of a random multiple-access system. *Problemy Peredachi Informatsii*, 23(3):64–78, 1987.