Scheduling in Networks with Time-Varying Channels and Reconfiguration Delay

Güner D. Çelik and Eytan Modiano
Laboratory For Information and Decision Systems
Massachusetts Institute of Technology

***Abstract*—We consider the optimal control problem for networks subjected to *time-varying channels*, *reconfiguration delays*, and *interference constraints*. We model the network by a graph consisting of nodes, links, and a set of link interference constraints, where based on the current network state, the controller decides either to stay with the current link-service configuration or switch to another service configuration at the cost of idling during schedule reconfiguration. Reconfiguration delay occurs in many telecommunications applications and is a new modeling component of this problem that has not been previously addressed. We show that the simultaneous presence of time-varying channels and reconfiguration delays significantly reduces the system stability region and changes the structure of optimal policies. We first consider memoryless channel processes and characterize the stability region in closed form. We prove that a frame-based Max-Weight scheduling algorithm that sets frame durations dynamically, as a function of the current queue sizes and *average* channel gains is throughput-optimal. Next, we consider arbitrary Markov modulated channel processes and show that memory in the channel processes can be exploited to improve the stability region. We develop a novel approach to characterizing the stability region of such systems using *state-action frequencies* which are stationary solutions to a Markov Decision Process (MDP) formulation. Finally, we develop a frame-based dynamic control policy, based on the state-action frequencies, and show that it is throughput-optimal asymptotically in the frame length. The FBDC policy is applicable to a broad class of network control systems, with or without reconfiguration delays, and provides *a new framework for developing throughput-optimal network control policies* using state-action frequencies.**

***Index Terms*—Reconfiguration delay, time-varying channels, scheduling, queueing, switching delay, Markov Decision Process**

I. Introduction

Scheduling in wireless networks subject to interference constraints has been studied extensively over the past two decades [8], [9], [12], [18], [19], [21]–[24]. However, to the best of our knowledge, the effects of *reconfiguration delays* have not been considered in the context of networks subject to *interference constraints* and *time-varying channel* conditions. Reconfiguration delay is a widespread phenomenon that is observed in many practical telecommunication systems [1], [5], [16], [27]. In satellite networks where mechanically steered antennas are providing service to ground stations, the time to switch from one station to another can be around 10ms [5], [25]. Similarly, in optical communication systems, laser tuning delay for transceivers can take significant time (μs-ms) [7], [16]. In wireless networks, delays for electronic beamforming or channel switching that occurs in phased-lock loops in oscillators can be more than $200\mu s$ [1], [5], [25], [27]. Worse yet, such small delay is often impossible to achieve due to delays incurred during different processing tasks such as channel estimation, signal to interference ratio calculation, inner loop closed power control, closed loop transmit diversity in the physical layer [1], [14], and stopping and restarting the interrupt service routines of various drivers in upper layers [1]. Moreover, in various real time implementations, channel switching delays from a few hundreds of microseconds to a few milliseconds have been observed [27].

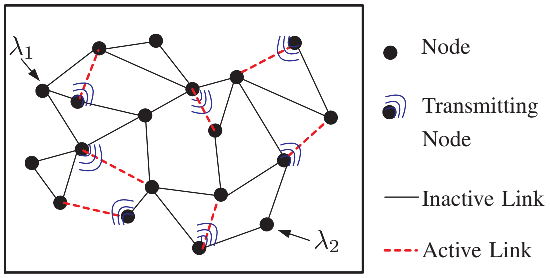

Fig. 1: System model. A single-hop wireless network with interference constraints, time-varying channels and reconfiguration delays.

We consider an optimal control problem for single-hop networks given by a graph structure $\mathcal{G}(\mathcal{N},\mathcal{L})$ of nodes $n \in \mathcal{N}$ and links $\ell \in \mathcal{L}$, subject to reconfiguration delays, time-varying channels, and arbitrary interference constraints. For the time-varying channel states, we consider both i.i.d. or Markov modulated processes for which the structure of the stability region and the optimal policies differ significantly. Our system model can be used to abstract single-hop wireless networks as show in Fig. 1, satellite networks with M servers and N ground stations as shown in Fig. 2, or NxN input-queued optical switches [21]. The network controller is to dynamically decide to stay with the current schedule of activations or to reconfigure to another schedule based on the channel process and the queue length information, where each decision to reconfigure leaves the network *idle* for a fixed amount of time, corresponding to *the reconfiguration delay*. Our goal is to study the impact of reconfiguration delays on system stability and optimal algorithms. We show that, as compared to systems without reconfiguration delays [18], [19], [23], [24], the stability region can be significantly reduced, and that optimal policies take on a different structure.

We first consider the case of memoryless (i.i.d.) channel

This work was supported by NSF grants CNS-0626781 and CNS-0915988, and by ARO Muri grant number W911NF-08-1-0238.

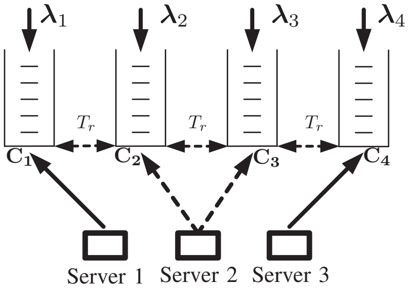

Fig. 2: An example 4x3 satellite network. Ground stations are subject to time-varying channels $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3$, $\mathbf{C}_4$ and the servers are subject to T_r slot reconfiguration (switchover) delay. Server 2 is forced to be idle due to interference constraints.

processes where we characterize the stability region in closed form as the convex hull of feasible activation vectors weighted by the *average* channel gain of each link. This result shows that in the presence of reconfiguration delays, it is not possible to take advantage of the diversity in time-varying channels because the i.i.d. channel processes refresh during each reconfiguration interval. Moreover, we show that a class of Variable-size Frame-based Max-Weight (VFMW) algorithms that make scheduling decisions based on time-average channel gains and queue lengths stabilize the system by keeping the current schedule over a frame of duration that is a function of the queue lengths.

Next, we consider Markov modulated channel processes with memory and develop *a novel methodology* to characterize the stability region of the system using *state-action frequencies*, the steady-state solutions to an MDP formulation for the corresponding saturated system. We show that the stability region enlarges with the memory in the channel processes, which is in contrast to the case of no reconfiguration delays [12], [19], [24]. Furthermore, we develop a novel frame-based dynamic control (FBDC) policy based on the state-action frequencies which achieves the full stability region. To our knowledge, this is the first throughout-optimal scheduling algorithm for wireless networks with time-varying channels and reconfiguration delays. The state-action frequency approach and the FBDC policy are applicable to many network control systems as they provide a general framework that reduces stability region characterization and throughput-optimal algorithm development to solving Linear Programs.

Optimal control of queuing systems and communication networks has been a very active research topic over the past two decades (e.g., [8], [12], [19], [23], [24]). In the seminal papers [23] and [24], Tassiulas and Ephremides characterized the stability region of multihop wireless networks and parallel queues with randomly varying connectivity. Later, these results were extended to power allocation and routing, scheduling under delayed or limited channel state information, and scheduling in switches (e.g., [7], [19], [21], [22]). However, these works do not consider the reconfiguration delays.

Switchover delay has been considered in Polling models in the Queuing Theory community (e.g., see [2] and the review in [26]), however, time-varying channels were not considered since they do not typically arise in classical Polling applications. Scheduling in optical networks under reconfiguration delay was considered in [7], [9], again in the absence of randomly varying connectivity. A detailed survey of the works in this field can be found in [26]. In [8] we considered a simple queuing system of two queues and a single server subject to ON/OFF channels and a single-slot switchover delay, where we developed the state-action frequency approach and the throughput-optimality of a frame-based policy. Here we generalize this model to arbitrary single-hop networks.

The main contribution of this paper is in solving the scheduling problem in single-hop networks under arbitrary *reconfiguration delays*, *time-varying channels* and *interference constraints* for the first time. We introduce the system model in detail in Section II. For systems with memoryless channel processes, we characterize the stability region and propose the class of throughput-optimal VFMW policies in Section III. We develop the state-action frequency approach and characterize the stability region for systems with Markov modulated channels in Section IV-A. We develop the throughput-optimal FBDC policy in Section IV-B and present simulation results in Section IV-C. For brevity, all proofs are omitted and can be found in our technical report [10].

II. Model

Consider a single-hop wireless network given by a graph structure $\mathcal{G}(\mathcal{N},\mathcal{L})$ of nodes $\mathcal{N}$ and links $\ell \in \mathcal{L} \doteq \{1, 2, ..., L\}$, where $L \doteq |\mathcal{L}|$. Data packets arriving at each link ℓ are to be transmitted to their single-hop destinations, where we refer to the packets waiting for service at link ℓ as queue ℓ. We consider a discrete-time (slotted) system where an integer number of data packets can arrive at or depart from the corresponding queue at each link during each time slot. Let the i.i.d. stochastic process $A_\ell(t)$ with arrival rate λ_ℓ denote the number of packets arriving to the source node of link ℓ at time slot t, where $\mathbb{E}[A_\ell^2(t)] \leq A_{\max}^2$, $\ell \in \mathcal{L}$. Let $\mathbf{Q}(t) = \{Q_1(t), ..., Q_L(t)\}$ denote the queue sizes at the links at the beginning of time slot t. Each link $\ell \in \mathcal{L}$ is subject to a time-varying channel process denoted by $C_\ell(t)$ that takes values in a set $\mathcal{C} = \{0, \mu_{\min}, ..., \mu_{\max}\}$ with $K \doteq |\mathcal{C}|$, where $C_\ell(t)$ corresponds to the number of packets that can be served from queue ℓ at time t. We consider both memoryless channel processes and Markovian channels with memory:

Definition 1 (Memoryless Channels): The channel process $\{C_\ell(t); t \geq 0\}$, $\ell \in \mathcal{L}$, takes independent and identically distributed (i.i.d.) values from the set $\mathcal{C}$ at each time slot t, according to a probability distribution for link ℓ, $\mathbb{P}^\ell$.

A simple example of a memoryless channel process is the Bernoulli process with 2-state i.i.d. ON-OFF channels.

Definition 2 (Channels With Memory): The channel process $\{C_\ell(t); t \geq 0\}$ forms the K-state irreducible and aperiodic Markov chain over the set $\mathcal{C}$, according to a transition probability distribution $\mathbb{P}^\ell(\cdot|j), j \in \mathcal{C}$.

The basic example of a Markovian channel process with memory is the commonly used Gilbert-Elliot channel model shown in Fig. 3. We let $\overline{C}_\ell$ denote the time-average channel

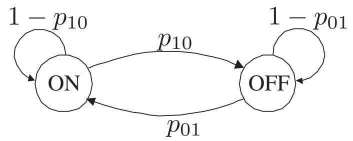

Fig. 3: Markov modulated ON/OFF channel process. The case of $p_{10}+p_{01} < 1$ provides positive correlation.

quality of link $\ell, \ell \in \mathcal{L}$, defined by

$$\overline{C}_\ell \doteq \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} C_\ell(\tau). \tag{1}$$

The limit exists both for memoryless and Markovian channel processes and is equal to the corresponding ensemble (steady state) average with probability (w.p.) 1 due to the Strong Law of Large Numbers (SLLN) [11]. We assume that all the arrival and channel processes, $A_\ell(t)$, $C_\ell(t), \ell \in \mathcal{L}$, are independent.

Let T_r denote the system reconfiguration delay, namely, it takes T_r time slots for the system to change a schedule, during which all the servers are necessarily idle. The set of all schedules in the system, $\mathcal{I}$, is given by the set of *feasible* binary activation vectors $\mathbf{I} = (\mathbf{I}_\ell)_{\ell=1,...,L}$, $\mathbf{I}_\ell \in \{0, 1\}$. If the activation vector $\mathbf{I}(t)$ is used at time slot t, then $\min\{C_\ell(t)\mathbf{I}_\ell(t), Q_\ell(t)\}$ packets depart from queue ℓ. We include the vectors dominated by the feasible activation vectors, as well as the zero vector $\mathbf{I} = \mathbf{0}$ in $\mathcal{I}$, where the activation vector $\mathbf{I}(t)$ is equal to $\mathbf{0}$ for all time slots at which the system is undergoing reconfiguration. A policy π is a mapping from the set of all possible queue length and action histories, to the set of all probability distributions on $\mathcal{I}$.

The availability of a schedule is determined by the interference constraints in the system, which are assumed to be arbitrary. For instance, in a wireless mesh network as shown in Fig. 1, the set $\mathcal{I}$ can be determined according to the well-studied k-hop interference model [12]. Alternatively, for a satellite network of N queues and M servers where there are a possible $L = NM$ links as shown in Fig. 2, the set $\mathcal{I}$ can be the set of all binary vectors of dimension NM with at most M nonzero elements such that no two active servers interfere with each other [9]. Finally, for an NxN input-queued optical switch, the set $\mathcal{I}$ can be the set of all matchings [21].

We say that an activation vector $\mathbf{I}$ is *ready to be activated in the current time slot* if the system does not need to reconfigure in order to activate $\mathbf{I}$, i.e., in such a case the servers that will be activated under $\mathbf{I}$ are *present* at their corresponding links at the beginning of the time slot. Finally, we assume that the queues are initially empty and that the arrivals take place after the departures in any given time slot. Under this model, the queue sizes evolve according to the following expression.

$$Q_\ell(t{+}1) = \max\{Q_\ell(t) - \mathbf{I}_\ell(t)C_\ell(t), 0\} + A_\ell(t), \forall \ell \in \mathcal{L}. \tag{2}$$

Definition 3 (Stability [17], [19]): The system is stable if

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{\ell\in\mathcal{L}} \mathbb{E}[Q_\ell(\tau)] < \infty.$$

For the case of integer valued arrival and departure processes as in this paper, this stability criterion implies the existence of a long-run stationary distribution for the queue size Markov chain with bounded first moments [17].

Definition 4 (Stability Region [17], [19]): The stability region $\mathbf{\Lambda}$ is the closure of the set of all arrival rate vectors $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_L)$ such that there exists a control algorithm that stabilizes the system.
The δ-stripped stability region is defined for some $\delta > 0$ as $\mathbf{\Lambda}^\delta \doteq \Big\{\boldsymbol{\lambda} | (\lambda_1 + \delta, ..., \lambda_L + \delta) \in \mathbf{\Lambda}\Big\}$.

Throughout the paper, we represent vectors, matrices, and sets with bold letters and we explicitly state when a variable is a matrix. We use the following notation for the inner product of two L-dimensional vectors: $\mathbf{u} \cdot \mathbf{v} \doteq \sum_{\ell=1}^{L} \mathbf{u}_\ell \mathbf{v}_\ell$.

III. Memoryless Channels

A. Stability Region

We start by characterizing the system stability region for the case of memoryless channels.

***Theorem** 1 (Stability Region $\mathbf{\Lambda}$ - Memoryless Channels): The stability region $\mathbf{\Lambda}$ is given by*

$$\mathbf{\Lambda} {=} \left\{\boldsymbol{\lambda} \middle| \exists\, \boldsymbol{\alpha} \geq \mathbf{0}, \sum_{\mathbf{I}\in\mathcal{I}} \alpha_{\mathbf{I}} \leq 1, \text{such that } \lambda_\ell \leq \overline{C}_\ell \sum_{\mathbf{I}\in\mathcal{I}} \alpha_{\mathbf{I}} \mathbf{I}_\ell, \forall \ell {\in} \mathcal{L}\right\}. \tag{3}$$

We prove the sufficiency of (3) in the next subsection where we show that for all $\boldsymbol{\lambda} \in \mathbf{\Lambda}$, a novel variable-size frame-based algorithm stabilizes the system. Here we highlight the basic ideas behind the necessity proof. Let $\mathbf{\Lambda}'$ denote the set at the right hand side of (3). This set can be rewritten as

$$\mathbf{\Lambda}' = \left\{\boldsymbol{\lambda} | \mathbf{C}^{-1}\boldsymbol{\lambda} \in Conv\{\mathcal{I}\}\right\}, \tag{4}$$

where $\mathbf{C}$ is a diagonal matrix with the ℓth diagonal element equal to $\overline{C}_\ell$, and $Conv\{\mathcal{I}\}$ denotes the convex hull of the set of all activations *and their descendants* given by

$$Conv\{\mathcal{I}\} \doteq \left\{\sum_{\mathbf{I}\in\mathcal{I}} \alpha_{\mathbf{I}} \mathbf{I} \;\middle|\; \text{for all } \boldsymbol{\alpha} \geq \mathbf{0} \text{ such that} \sum_{\mathbf{I}\in\mathcal{I}} \alpha_{\mathbf{I}} = 1\right\}.$$

Fix $\boldsymbol{\lambda} \notin \mathbf{\Lambda}'$. We show that $\mathbf{\Lambda} \subseteq \mathbf{\Lambda}'$ by proving that $\boldsymbol{\lambda} \notin \mathbf{\Lambda}$. From the closed convex set separation theorem [4], there exist a hyperplane, i.e., a vector $\mathbf{h} \geq \mathbf{0}$ and a constant $b > 0$, such that

$$\boldsymbol{\lambda} \cdot \mathbf{h} > b > (\mathbf{C}\boldsymbol{\mu}) \cdot \mathbf{h}, \;\; \forall \boldsymbol{\mu} \in Conv\{\mathcal{I}\}. \tag{5}$$

For contradiction, suppose that there exists a stabilizing policy $\boldsymbol{\pi}$. The following expression, similar to (2), holds for all links:

$$Q_\ell(t) \geq Q_\ell(0) + \sum_{\tau=0}^{t-1} A_\ell(\tau) - \sum_{\tau=0}^{t-1} \mathbf{I}_\ell(\tau) C_\ell(\tau). \tag{6}$$

Consider (6) for a link ℓ with $\lambda_\ell > 0$. Let $\mu_\ell(t) \doteq \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbf{I}_\ell(\tau)C_\ell(\tau)$. Let $\hat{\mathbf{I}}(t)$ be the activation vector that is *ready to be activated* at the beginning of time slot t under policy $\boldsymbol{\pi}$. Note that the vectors $\hat{\mathbf{I}}(t)$ and $\mathbf{I}(t)$ are different only if t is the first time slot of a reconfiguration interval. Let the counting process $M_\ell(t)$ be the number of time slots between times 0 and t for which an activation vector $\hat{\mathbf{I}}$ that activates link ℓ is *ready to be activated* under policy $\boldsymbol{\pi}$, and let $t_0, t_1, ..., t_{M_\ell(t)}$ be these time slots. By definition we have $M_\ell(t) = \sum_{\tau=0}^{t-1} \hat{\mathbf{I}}_\ell(t)$. Note that $\lim_{t\to\infty} M_\ell(t) = \infty$, because if $\lim_{t\to\infty} M_\ell(t) < B$ for some constant B, then the time-average service rate of link ℓ is 0 and queue ℓ is unstable, a contradiction. We have

$$\begin{aligned} \mu_\ell(t) &= \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbf{I}_\ell(\tau)C_\ell(\tau) \le \frac{1}{t}\sum_{\tau=t_0}^{t_{M_\ell(t)}} \hat{\mathbf{I}}_\ell(\tau)C_\ell(\tau), \\ &\le \frac{M_\ell(t)}{t}\frac{1}{M_\ell(t)}\sum_{\tau=t_0}^{t_{M_\ell(t)}} C_\ell(\tau). \qquad (7) \end{aligned}$$

Since $M_\ell(t) \to \infty$ in the infinite time horizon, the infinite sequence $\{C_\ell(\tau)\}_{\tau\in\{t_0,t_1,t_2,...\}}$ forms an i.i.d. sequence. Therefore, we have from the SLLN [11] that $\lim_{t\to\infty} \frac{1}{M_\ell(t)} \sum_{\tau=t_0}^{t_{M_\ell(t)}} C_\ell(\tau) = \overline{C}_\ell$. Hence, for some large time t, $\mu_\ell(t)$ is at most about $\overline{C}_\ell M_\ell(t)/t$, where $\frac{M_\ell(t)}{t} = \frac{1}{t}\sum_{\tau=0}^{t-1} \hat{\mathbf{I}}_\ell(t)$ belongs to $Conv\{\mathcal{I}\}$ for all t [12]. Now for $\boldsymbol{\lambda} \notin \boldsymbol{\Lambda}'$, we utilize the convex set separation theorem as in (5) to obtain that $(\boldsymbol{\lambda} - \mathbf{C}(\frac{1}{t}\sum_{\tau=0}^{t-1} \hat{\mathbf{I}}_\ell(t))) \cdot \mathbf{h} > 0$. The rest of the proof is based on the fact that this creates a positive drift for the queue sizes upon taking expectation of (6).

Theorem 1 shows that in the presence of reconfiguration delays, no policy can take advantage of the diversity in time-varying memoryless channels and achieve a greater rate than the average channel gain for each link. This is because the system cannot switch to another schedule instantly in order to opportunistically exploit better channel states and observes an average channel gain upon switching. This is in sharp contrast to the corresponding systems without reconfiguration delay considered in [18], [19], where throughput-optimal policies are able to take advantage of the diversity in i.i.d. channels by instantly and opportunistically switching schedules. As shown in the example below for a simple two-queue system, this negative impact of the reconfiguration delay reduces the stability region considerably as compared to systems without reconfiguration delays [19].

Theorem 1 also establishes that the duration of the reconfiguration interval has no effect on the stability region for memoryless channel processes. This is because for memoryless channels, giving infrequent reconfiguration decisions minimizes the fraction of time slots lost to reconfiguration.

1) Example: Two Queues and a Single Server: Consider i.i.d. ON/OFF channels with probability of ON channel state equal to 0.5 for both queues for all time slots, and 1 slot switchover delay for the server. The stability region of this system can be obtained from (3) and it takes a simple structure:

$$\boldsymbol{\Lambda} = \{(\lambda_1, \lambda_2) \big| \lambda_1 + \lambda_2 \le 0.5, \lambda_1, \lambda_2 \ge 0\}. \qquad (8)$$

The stability region of the corresponding system with no switchover delay was established in [24]: $\lambda_1, \lambda_2 \in [0, 0.5]$ and $\lambda_1 + \lambda_2 \le 0.75$. As depicted in Fig. 4, even 1-slot switchover delay reduces the stability region of the system considerably. Note that for systems without time-varying channels, stability conditions are given by $\lambda_1 + \lambda_2 \le 1, \lambda_1, \lambda_2 \ge 0$ and is *not* affected by the switchover delay [26]. Therefore, *it is the combination of switchover delay and time-varying channels that results in fundamental changes in system stability*. As we show in Section IV, and as displayed in Fig. 4, the memory in the channel processes can be exploited to improve the system stability region when the reconfiguration delay is nonzero.

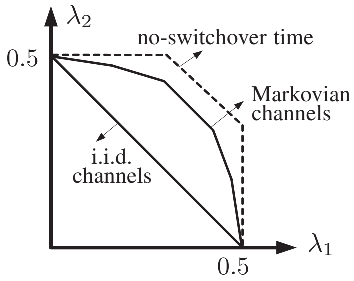

Fig. 4: Stability region under memoryless (i.i.d.) channels and channels with memory (Markovian with $p_{10} + p_{01} < 1$) with and without switchover delay.

B. Variable Frame Based Max-Weight (VFMW) Algorithm

In this section we propose a throughput-optimal algorithm based on the following intuition: Given that no policy can take advantage of the diversity in channel processes, giving infrequent reconfiguration decisions minimizes throughput lost to reconfiguration. For networks with nonzero reconfiguration delays, in the absence of randomly varying connectivity, we proved in [9] that a variable-size frame-based Max-Weight algorithm which keeps the same schedule over a frame of duration based on the queue lengths is throughput-optimal. We show here that an adaptation of the algorithm in [9] that also takes into account *the average channel gains of time-varying links* is throughput-optimal for systems with memoryless channel processes. Specifically, let t_k be the first slot of the kth frame, let $\mathbf{Q}(t_k)$ be the queue lengths at t_k, and let $S(\mathbf{Q}(t_k)) \doteq \sum_i Q_i(t_k)$. The VFMW policy calculates the Max-Weight schedule with respect to $\mathbf{Q}(t_k)$ and $\overline{\mathbf{C}} \doteq (\overline{C}_1, ..., \overline{C}_L)$, and applies this schedule during the frame as defined in detail in Algorithm 1.

Algorithm 1 VFMW ALGORITHM

```
1: Find the Max-Weight schedule at time $t_k$, $\mathbf{I}^*(t_k)$, w.r.t. $\mathbf{Q}(t_k)$ weighted by the average channel gains $\overline{\mathbf{C}}$:
       $\mathbf{I}^*(t_k) = \arg\max_{\mathbf{I}\in\mathcal{I}} \sum_\ell \mathbf{I}_\ell \overline{C}_\ell Q_\ell(t_k)$
2: If $\mathbf{I}^*(t_k) \neq \mathbf{I}^*(t_{k-1})$, then invoke reconfiguration for the next $T_r$ slots.
3: Apply $\mathbf{I}^*(t_k)$ for an interval of duration $F(S(\mathbf{Q}(t_k)))$ slots where $\chi_k \doteq T_r + F(S(\mathbf{Q}(t_k)))$, $F(\cdot) > 0$ is a monotonically increasing sublinear function, i.e., $\lim_{y\to\infty} F(y)/y = 0$.
4: Repeat above for the next frame starting at $t_{k+1} = t_k + \chi_k$.
```

The VFMW algorithm sets the frame length as a suitably increasing sublinear function of the queue lenghts, which dynamically adapts the frame duration to the stochastic arrivals.

For instance, $\chi_k = T_r + (\sum_i Q_i(t_k))^{\alpha}$ with $\alpha \in (0,1)$ satisfies the criteria for the frame duration. Under the VFMW policy, the frequency of service reconfiguration is small when the queue lengths are large, limiting the fraction of time spent to switching. Note that this frequency should not be too small otherwise the system becomes unstable as it is subjected to a bad schedule for an extended period of time. When the queue lenghts are small, the VFMW policy gives frequent reconfiguration decisions, becoming more adaptive and providing good delay performance.

***Theorem** 2: The VFMW policy stabilizes the system for all arrival rates $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}$, without requiring knowledge of $\boldsymbol{\lambda}$.*

An immediate corollary to this theorem is as follows:

***Corollary** 1: The conditions in (3) are sufficient for stability.*

The proof establishes the fact that the drift over the switching epochs, i.e., $\mathbb{E}\big[L(\mathbf{Q}(t_k+\chi_k)) - L(\mathbf{Q}(t_k))\big|\mathbf{Q}(t_k)\big]$, is negative using the quadratic Lyapunov function, $L(\mathbf{Q}(t)) = \sum_{\ell=1}^{L} Q_\ell^2(t)$. The basic intuition behind the proof is that if the queue sizes are large, the VFMW policy accumulates sufficient negative drift during the frame, which overcomes the cost accumulated during reconfiguration. Moreover, for large queue lengths, since the policy keeps the same schedule during the resulting long frames, we obtain the time-average channel gains in the system, as seen in the stability condition in (4). Note that choosing the frame length as a *sublinear* function of the queue sizes is critical. This is because the VFMW algorithm uses the Max-Weight schedule corresponding to the beginning of the frame, which "loses weight" as the frame goes on. Therefore, one needs to make sure that the system is not subjected to this "light-weight" schedule for too long. In particular, frame lengths sublinear in queue sizes work, however, frame lengths that are linear in queue sizes do not guarantee stability. This is because under linear switching curves, the drift of the queue lengths over the frames becomes positive under commonly used polynomial Lyapunov functions $L(\cdot)$ of power greater than 1.

In Section IV, we show that for channel processes with memory, delaying the reconfiguration decisions as in VFMW algorithm does not work and more sophisticated algorithms are necessary in order to exploit the channel memory.

C. Simulation Results - Memoryless Channels

We performed simulation experiments that determine average queue occupancy values for the VFMW policy, the ordinary Max-Weight (MW) policy and the Max-Weight policy with fixed frame sizes (FFMW), where the MW policy "chooses" the schedule $\arg\max_{\mathbf{I}} \sum_\ell Q_\ell(t) C_\ell(t) \mathbf{I}_\ell$, and the FFMW policy applies the same activation vector as the VFMW policy over frames of constant duration. The average total queue occupancy over T_s slots is defined by $\frac{1}{T_s}\sum_{t=1}^{T_s}\sum_{\ell\in\mathcal{L}} Q_\ell(t)$ and the frame length for the VFMW policy is chosen as $\chi_k = T_r + (\sum_i Q_i(t_k))^{0.9}$. Through Little's law, the long-run packet-average delay in the system is equal to the time-average number of packets divided by the total arrival rate into the system. We considered a network of 4 links and 3 servers as shown in Fig. 2, where server 1 and 3 are dedicated to links (queues) 1 and 4 respectively, and server 2 is shared between queues 2 and 3. Due to interference constraints, no two links that are "adjacent" to each other can be activated simultaneously, namely, the feasible activation vectors are given by $\mathbf{I}^1 = [1010]$, $\mathbf{I}^2 = [0101]$, and $\mathbf{I}^3 = [1001]$. For each data point, the simulation length was 100,000 slots, and the arrival and the channel processes were i.i.d. Bernoulli, with arrival rate λ, and probability of ON channel state equal to 0.5, respectively.

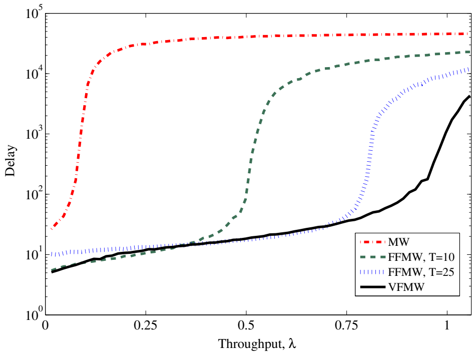

Fig. 5: Delay vs throughput under the VFMW, MW, and the FFMW policies.

We simulated delay as a function of sum-throughput $\sum_\ell \lambda_\ell$ for $\boldsymbol{\lambda}$ along the line between the origin and the maximum sum-throughput point $\boldsymbol{\lambda}_{\max}$ given by $\arg\max_{\boldsymbol{\lambda}\in\boldsymbol{\Lambda}} \sum_\ell \lambda_\ell$, where from (3), $\boldsymbol{\lambda}_{\max}$ can be calculated to be $[0.33\ 0.17\ 0.17\ 0.33]$ with $\sum_\ell \lambda_\ell = 1$. Note that maximum sum-throughput for the corresponding system with zero reconfiguration delay is about 1.44 [12], which shows the significant reduction in throughput due to the reconfiguration delay. Fig. 5 presents the delay as a function of sum-throughput for the VFMW, MW, and the FFMW (with frame sizes $T{=}10$ and $T{=}25$) policies, for $T_r{=}5$ slot reconfiguration delay. Fig. 5 confirms that as the arrival rates are increased, the system quickly becomes unstable under the MW policy and that the VFMW policy provides stability for all sum-rates less than 1. The FFMW policy has larger stability region than that of the MW policy, and increasing the frame length of the FFMW policy improves it's stability region at the expense of delay performance. The VFMW policy provides a good balance by dynamically adapting the frame length based on the queue sizes and stabilizes the system whenever possible, while providing a delay performance that is similar to that of a FFMW policy with a small frame length for small arrival rates.

IV. Channels With Memory

In this section we establish the stability region of the system and propose a throughput-optimal dynamic control policy when the time-varying channels have memory. We generalize the novel framework of characterizing the stability region in terms of state-action frequencies that we introduced in [8] to wireless networks with reconfiguration delays, time-varying channels, and interference constraints. The state-action frequency approach is a general and unifying framework in that, for the simpler case of no-reconfiguration delay in the system, it provides the stability region characterizations of classical network control papers such as [18], [21], [24].

We show that *the stability region expands with memory in the channel processes*, in particular, it lies between the stability region for the case of i.i.d. channels and the stability region for the case of no reconfiguration delay. For classical network control systems such as [18], [19], [24], the memory in the channel processes does not affect the stability region [12]. Therefore, scheduling under reconfiguration delays and time-varying channels calls for new control algorithms that can improve their performance with increases in channel memory.

A. Stability Region

We start by analyzing the corresponding system with saturated queues, i.e., all queues are always non-empty. Let $\mathbf{\Lambda}_s$ denote the set of all time average expected departure rate vectors $\mathbf{r} = (r_1, ..., r_L)$ that can be obtained in the saturated system under all possible policies that are possibly history dependent, randomized, or non-stationary. We will show that the stability region $\mathbf{\Lambda}$ satisfies $\mathbf{\Lambda} = \mathbf{\Lambda}_s$. We show the necessary stability conditions in the following Lemma and establish sufficiency in the next section.

Lemma 1: *We have that* $\mathbf{\Lambda} \subseteq \mathbf{\Lambda}_s$.

An intuitive explanation for the proof is that given a policy $\boldsymbol{\pi}$ for the saturated system, we apply the same policy in the system with random arrivals with *the same sample path of channel realizations*. It is clear that the total number of departures from each queue in the saturated system is no less than that in the original system. We establish the region $\mathbf{\Lambda}_s$ by formulating the system dynamics as a Markov Decision Process (MDP).

1) MDP Formulation For Saturated System: We present the analysis for the case of a single slot reconfiguration delay, i.e., $T_r = 1$, for ease of exposition and we demonstrate how to generalize the analysis to the case of $T_r > 1$ whenever appropriate[1]. For $T_r = 1$, let $\mathbf{s}_t = (\mathbf{I}(t), \mathbf{C}(t)) \in \mathcal{S}$ denote the system state at time t, where $\mathbf{C}(t)$ is the vector of channel processes at each link at time slot t, and $\mathcal{S}$ is the set of all states. Also, let $\mathbf{a}_t \in \mathcal{I}$ denote the action taken at time slot t, which determines the activation vector that will be available at the beginning of the next time slot. For $T_r > 1$ the state has one more variable that represents how far into the reconfiguration interval the system is.

Let $\mathbb{H}(t) = [\mathbf{s}_\tau]|_{\tau=0}^{t} \cup [\mathbf{a}_\tau]|_{\tau=0}^{t-1}$ denote the full history of the system until time t and let $\Upsilon(\mathcal{I})$ denote the set of all probability distributions on $\mathcal{I}$. For the saturated system, a policy is a mapping from $\mathbb{H}(t)$ to $\Upsilon(\mathcal{I})$ [3], [15], [20]. Namely, a policy prescribes the probability of any particular action for a given system history. A *stationary* policy is a policy that depends only on the current state, and under a stationary policy the process $\{\mathbf{s}_t; t \geq 0\}$ forms a Markov chain. For a stationary policy, the server observes the current state $\mathbf{s}_t$ and chooses an action a_t. Then the next state j is realized according to the transition probabilities $\mathbb{P}(j|\mathbf{s}, \mathbf{a})$, which depend on the random channel processes. Now, we define the rewards as a function of the state $\mathbf{s} = (\mathbf{I}, \mathbf{C})$ as follows:

$$\overline{r}_l(\mathbf{s}, \mathbf{a}) \doteq C_l, \text{ if } \mathbf{I}_\ell = 1 \text{ and } \mathbf{a}{=}\mathbf{I}, \tag{9}$$

and $\overline{r}_\ell(\mathbf{s}, \mathbf{a}) \doteq 0$ otherwise. That is, a reward of C_ℓ is obtained if the controller decides to stay with the current schedule and if link ℓ is active under the current schedule. We are interested in the set of all possible time average expected departure rates. Therefore, given some $\alpha_\ell \geq 0$, $\ell \in \mathcal{L}$, $\sum_\ell \alpha_\ell = 1$, we define the system reward at time t by the weighted sum-rate $\overline{r}(\mathbf{s}_t, \mathbf{a}_t) \doteq \sum_{\ell \in \mathcal{L}} \alpha_\ell \overline{r}_\ell(\mathbf{s}_t, \mathbf{a}_t)$. The average reward of policy $\boldsymbol{\pi}$ is defined by

$$r^{\boldsymbol{\pi}} \doteq \limsup_{K \to \infty} \frac{1}{T} E\Big[\sum_{t=1}^{T} \overline{r}(\mathbf{s}_t, \mathbf{a}_t^{\boldsymbol{\pi}})\Big]. \tag{10}$$

Given weights $\alpha_\ell, \ell{\in}\mathcal{L}$, we are interested in the policy that achieves the maximum time average expected reward $r^* \doteq \max_{\boldsymbol{\pi}} r^{\boldsymbol{\pi}}$. This optimization problem is a discrete time MDP characterized by the state transition probabilities $\mathbb{P}(.|\mathbf{s}, \mathbf{a})$ with $K^L|\mathcal{I}|$ states and $|\mathcal{I}|$ actions per state, where K is the number of channel states. Furthermore, any given pair of states are accessible from each other (i.e., there is a positive probability path between the states) under some stationary deterministic policy. Therefore, this MDP belongs to the class of *Weakly Communicating* MDPs [20], for which there exists a stationary deterministic optimal policy independent of the initial state, given as a solution to the standard Bellman's equation [20].

2) State-Action Frequency Approach: The *state-action frequency* approach, or the *Dual Linear Program (LP)* approach, given below provides a systematic and intuitive framework to solve such average cost MDPs, and it can be derived using Bellman's equation and the monotonicity property of Dynamic Programs [Section 8.8] [20]:

$$\text{Maximize} \quad \sum_{\mathbf{s} \in \mathcal{S}} \sum_{\mathbf{a} \in \mathcal{I}} \overline{r}(\mathbf{s}, \mathbf{a}) \mathbf{x}(\mathbf{s}, \mathbf{a})$$

subject to the balance equations

$$\sum_{\mathbf{a} \in \mathcal{I}} \mathbf{x}(\mathbf{s}, \mathbf{a}) = \sum_{\mathbf{s}' \in \mathcal{S}} \sum_{\mathbf{a} \in \mathcal{I}} \mathbb{P}\big(\mathbf{s}|\mathbf{s}', \mathbf{a}\big) \mathbf{x}(\mathbf{s}', \mathbf{a}), \ \forall \ \mathbf{s} \in \mathcal{S}, \tag{11}$$

the normalization condition $\sum_{\mathbf{s} \in \mathcal{S}} \sum_{\mathbf{a} \in \mathcal{I}} \mathbf{x}(\mathbf{s}, \mathbf{a}){=}1$, and the nonnegativity constraints $\mathbf{x}(\mathbf{s}, \mathbf{a}){\geq}0$, for $\mathbf{s} \in \mathcal{S}, \mathbf{a} \in \mathcal{I}$, where the transition probabilities $\mathbb{P}\big(\mathbf{s}|\mathbf{s}', \mathbf{a}\big)$ are functions of the channel transition probabilities. The feasible region of this LP constitutes a polytope called the *state-action polytope* $\mathbf{X}$ and the elements of this polytope $\mathbf{x} \in \mathbf{X}$ are called state-action frequency vectors. It can be shown that $\mathbf{X}$ is convex, bounded, and closed. Note that a point $\mathbf{x} \in \mathbf{X}$ corresponds to a stationary randomized policy that takes action $\mathbf{a} \in \mathcal{I}$ at state $\mathbf{s}$ w.p.

$$\mathbb{P}(\textit{action } \mathbf{a} \text{ at state } \mathbf{s}) {=} \frac{\mathbf{x}(\mathbf{s}, \mathbf{a})}{\sum_{\mathbf{a}' \in \mathcal{I}} \mathbf{x}(\mathbf{s}, \mathbf{a}')}, \mathbf{a} \in \mathcal{I}, \mathbf{s} \in S_x, \tag{12}$$

where S_x is the set of recurrent states given by $S_x \equiv \{\mathbf{s} \in S : \sum_{\mathbf{a} \in \mathcal{I}} \mathbf{x}(\mathbf{s}; \mathbf{a}) > 0\}$ [15], [20]. For transient states $\mathbf{s} \in S/S_x$,

[1] In a slotted system, even a minimal reconfiguration delay will lead to a loss of a slot due to synchronization issues.

an action that leads the system to a recurrent state is chosen. Every point $\mathbf{x}$ at an interior or a corner point of $\mathbf{X}$ can be achieved by a stationary randomized policy as defined in (12) [15], [20]. An inverse statement also holds, namely, the expected empirical state-action frequency vector of *any* policy lies in $\mathbf{X}$ regardless of the initial state distribution. The following lemma establishes the equivalency between the corners of the state-action polytope $\mathbf{X}$ and stationary deterministic policies [15], [20].

***Lemma** 2: The vertexes of the LP in (11) have a one-to-one correspondence with stationary-deterministic policies.*
If $\mathbf{x}$ is an extreme point of $\mathbf{X}$, it cannot be expressed as a convex combination of any two other elements in $\mathbf{X}$, therefore, for each state $\mathbf{s}$ only one action has a nonzero probability.

3) The Rate Polytope $\mathbf{\Lambda}_s$: Using the theory of state-action polytopes in the previous section, we characterize the set of all achievable time-average expected rates in the saturated system, $\mathbf{\Lambda}_s$. The following linear transformation of the state-action polytope $\mathbf{X}$ defines the L dimensional *rate polytope* [15]:

$$\mathbf{\Lambda}_s = \Big\{\mathbf{r}\Big| r_\ell = \sum_{\mathbf{s}\in\mathcal{S}}\sum_{\mathbf{a}\in\mathcal{I}} \mathbf{x}(\mathbf{s},\mathbf{a})\overline{r}_\ell(\mathbf{s},\mathbf{a}), \ell \in \mathcal{L}\Big\}, \tag{13}$$

where $\overline{r}_\ell$ is the reward function for link ℓ defined in (9). This polytope is the set of all time average expected departure rate pairs that can be obtained in the saturated system, i.e., it is the rate region $\mathbf{\Lambda}_s$. Furthermore, because $\mathbf{\Lambda}_s$ is a linear transformation of $\mathbf{X}$, corner points of $\mathbf{\Lambda}_s$ are also achieved by stationary deterministic policies. An explicit way of characterizing $\mathbf{\Lambda}_s$ is given in Algorithm 2. Note that (14) is an LP because $r_\ell(\mathbf{x}), \ell \in \mathcal{L}$, are linear functions of $\mathbf{x}(\mathbf{s},\mathbf{a})$ defined through (9), where $\mathbf{s} = (\mathbf{I},\mathbf{C})$. This is because $\overline{r}_\ell$ is a linear combination of those $\mathbf{x}(\mathbf{s},\mathbf{a})$ vectors for which the action $\mathbf{a}$ is the same as the current schedule in the state $\mathbf{I}$, the lth component of $\mathbf{I}$ is 1, and C_ℓ is the coefficient in the linear combination. The fundamental theorem of Linear Programming guarantees existence of an optimal solution to (14) at a corner point of the polytope $\mathbf{X}$ and hence of $\mathbf{\Lambda}_s$ [6]. We will establish in the next section that the rate region $\mathbf{\Lambda}_s$ is in fact achievable in the dynamic queueing system, which will imply that $\mathbf{\Lambda} = \mathbf{\Lambda}_s$. For the simple two-queue system introduced in Section III-A1, the structure of the rate polytope in (14) is analyzed in detail in [8]. Furthermore, the one-to-one correspondence between the extreme points of the polytope $\mathbf{X}$ and stationary deterministic polices stated in Lemma 2 is useful for finding the solutions of the above LP. For instance, for the two-queue system, the LP in (14) can be solved explicitly for all $\boldsymbol{\alpha}$ values to derive the rate region $\mathbf{\Lambda}_s$, which is displayed in Fig. 6. As expected, the stability region is improved for channels with memory as compared to the stability region for the case of i.i.d. channels shown in Fig. 4. Moreover, the region $\mathbf{\Lambda}_s$ is significantly reduced as compared to the stability region for the corresponding system with zero switchover delays shown with dashed lines in Fig. 6 [24].

Algorithm 2 *Stability Region Characterization*

```
1: Given $\alpha_1, ..., \alpha_L \geq 0$, $\sum_\ell \alpha_\ell = 1$, solve the following LP
       $\max_{\mathbf{x}}. \quad \sum_{\ell=1}^{L} \alpha_\ell r_\ell(\mathbf{x})$
       subject to $\quad \mathbf{x} \in \mathbf{X}$.                    (14)
2: There exists an optimal solution $(r_1^*, ..., r_L^*)$ to this LP
   that lies at a corner point of $\mathbf{\Lambda_s}$ and hence of $\mathbf{X}$. Find all
   possible corner points and take their convex combination.
```

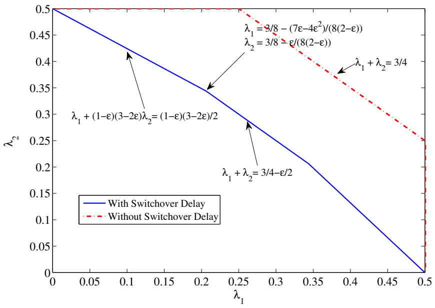

Fig. 6: Stability region for the simple system with and without switchover time for Gilbert-Elliot channel model with $p_{10} = p_{01} = 0.40$.

B. Frame Based Dynamic Control Policy

We propose a frame-based dynamic control (FBDC) policy inspired by the state-action frequency approach and prove that it is throughput-optimal asymptotically in the frame length. The motivation behind the FBDC policy is that a policy $\boldsymbol{\pi}^*$ that achieves the optimization in (14) for given weights $\alpha_\ell, \ell \in \mathcal{L}$, for the saturated system should achieve a *good* performance in the original system when the queue sizes $\mathbf{Q}$ are used as weights. This is because first, the policy $\boldsymbol{\pi}^*$ will lead to similar average departure rates in both systems for sufficiently large queue lengths, and second, the usage of queue lengths as weights creates self adjusting policies that dynamically capture the changes due to stochastic arrivals, similar to the Max-Weight scheduling algorithm. Specifically, we divide the time into equal-size intervals of T slots and let $\overline{\mathbf{Q}}(jT)$ denote the queue lengths at the beginning of the jth interval, normalized by $\sum_\ell Q_\ell(jT)$. We find the stationary deterministic policy that optimally solves (14) when $\overline{Q}_1(jT), ..., \overline{Q}_L(jT)$ are used as weights and then apply this policy throughout the frame in the dynamic queueing system. The FBDC policy is described in detail in Algorithm 3. The LP in (15) can be restated as $\max_{\{\mathbf{r}\}} \sum_{\ell=1}^{L} \overline{Q}_\ell(jT) r_\ell$ subject to $\mathbf{r} \in \mathbf{\Lambda}_s$. There exists an optimal solution $\mathbf{r}^*$ of the LP in (15) that is a corner point of $\mathbf{X}$ (and hence of $\mathbf{\Lambda}_s$) [6], and the policy $\boldsymbol{\pi}^*$ that corresponds

Algorithm 3 Frame Based Dynamic Control Policy

```
1: Find the policy $\boldsymbol{\pi}^*$ that optimally solves the following LP
       $\max_{\mathbf{x}}. \quad \sum_{\ell=1}^{L} \overline{Q}_\ell(jT) r_\ell(\mathbf{x})$
       subject to $\quad \mathbf{x} \in \mathbf{X}$                    (15)
2: Apply $\boldsymbol{\pi}^*$ in each time slot of the frame.
```

to this point is a stationary deterministic policy by Lemma 2.

***Theorem** 3: For any $\delta > 0$, there exists a large enough frame length T such that the FBDC policy stabilizes the system for all arrival rates within the δ-stripped stability region $\mathbf{\Lambda}_s^\delta = \mathbf{\Lambda}_s - \delta\mathbf{1}$.*

This theorem immediately implies that $\mathbf{\Lambda} = \mathbf{\Lambda_s}$. The proof performs a drift analysis using the standard quadratic Lyapunov function. However, *it is novel in utilizing the MDP framework within the Lyapunov drift arguments*. The basic idea is that, for sufficiently large queue lengths, when the optimal policy solving (15), $\boldsymbol{\pi}^*$, is applied over a sufficiently long frame of T slots, the average output rates of both the actual system and the corresponding saturated system converge to $\mathbf{r}^*$. For the saturated system, the probability of a large difference between empirical and steady state rates, captured by the parameter δ, decreases exponentially fast in T [15], similar to the convergence of a positive recurrent Markov chain to its steady state. Therefore, for sufficiently large queue lengths, the difference between the empirical rates in the actual system and $\mathbf{r}^*$ also decreases with T. This ultimately results in a negative Lyapunov drift since $\mathbf{r}^*$ is the solution to $\max_{\mathbf{r}} \sum_{\ell=1}^{L} \overline{Q}_\ell(jT) r_\ell$, $\mathbf{r} \in \mathbf{\Lambda}_s$, leading to $\sum_\ell Q_\ell(jT) r_\ell^* \geq \sum_\ell Q_\ell(jT)\lambda_\ell$, for all $\boldsymbol{\lambda}$ in $\mathbf{\Lambda}_s$.

The FBDC policy implemented without any frames, i.e., for $T = 1$, has a similar performance to the original FBDC policy. This is because for large queue lengths, the optimal solution of the LP in (15) depends on the queue length ratios, and hence, the policy $\boldsymbol{\pi}^*$ that solves the LP optimally does not change fast when the queue lengths get large.

***Remark** 1:* The FBDC policy provides *a new framework for developing throughput-optimal policies for network control*. Namely, given any queuing system whose corresponding saturated system is Markovian with finite state and action spaces, throughput-optimality is achieved by solving an LP in order to find the stationary MDP solution for the corresponding saturated system and applying this solution over frames in the actual system. The FBDC policy can also be used to achieve throughput-optimality for classical network control systems [18], [24], optical switches [21], or systems with delayed channel state information [22].

It is well-known that the celebrated Max-Weight scheduling policy is not throughput-optimal for systems with reconfiguration delays [9]. In the absence of time-varying channels, or for systems with i.i.d. channel processes as in Section III, variable frame-based generalizations of the Max-Weight policy are throughput-optimal [9]. However, under the simultaneous presence of reconfiguration delays and time-varying channels with memory in a network, the FBDC policy is the only policy to achieve throughput-optimality and it has a significantly different structure from the Max-Weight policy.

The state space of the LP needed to be solved under the FBDC policy increases exponentially with the number of links in the system. For a single-hop network with interference constraints, the Max-Weight policy has to solve a maximum-independent set problem over all links at each time slot, which is a hard problem whose state space also increases exponentially in the number of links. The FBDC policy, on the other hand, only has to solve an LP, for which there are standard solvers available such as CPLEX. Furthermore, the FBDC policy has to solve the LP once per frame, whereas the the Max-Weight policy performs maximum-independent set computation each time slot. If the frame length for the FBDC policy is chosen to be bigger than the computational complexity of the LP in (15), then the per-slot computational complexity of the algorithm is reduced to $O(1)$. Finally, the FBDC policy requires the knowledge of the channel process statistics. To deal with this problem, one can estimate the channel parameters periodically and use these estimates to solve the LP in (15).

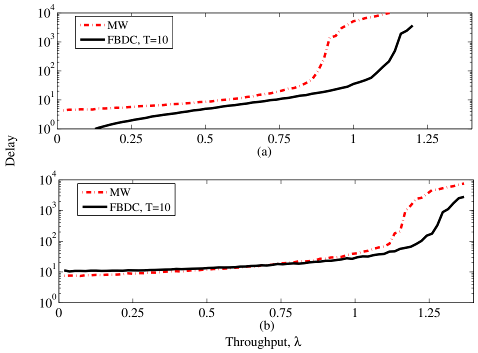

Fig. 7: Delay vs the throughput under the FBDC policy with frame length 10 and MW policies, (a) for $p_{10} = p_{01} = 0.25$, and (b) for $p_{10} = p_{01} = 0.10$.

C. Simulation Results - Channels with Memory

We performed simulation experiments that determine average queue occupancy values for the FBDC policy and the ordinary Max-Weight (MW) policy. We consider the same simulation model as in Section III-C, except that we have Gilbert-Elliot channel model with two sets of transition probabilities $p_{10} = p_{01} = 0.10$ and $p_{10} = p_{01} = 0.25$ for each channel and that the switchover delay T_r is taken to be 1 slot. As for the case of i.i.d. channels considered in Section III-C, the steady state probability of ON channel state for each queue is 0.5 under both sets of transition probabilities. As we demonstrate in this section, the stability region is larger for the case of channels with memory. This is in sharp contrast to systems with zero reconfiguration delays for which the stability region only depends on the steady state behavior of the channel processes [12]. Furthermore, the maximum sum-throughput for this system can be calculated to be 1.14 for $p_{10} = p_{01} = 0.25$, and 1.30 for $p_{10} = p_{01} = 0.10$ by numerically solving the LP in (14) for weights $\alpha_i = \alpha_j, i, j \in \mathcal{L}$. While these values are significantly larger than the maximum sum-throughput of 1 for the case of i.i.d. channels considered in Section III-C, they are less than the sum-throughput of 1.44 for the corresponding system with zero reconfiguration delays, as expected.

Fig. 7(a) presents delay as a function of sum-throughput along the line between the origin and the maximum sum-throughput point for the FBDC policy with frame length 10 and the MW policy for $p_{10} = p_{01} = 0.25$. This figure shows that the system becomes unstable around the sum-throughput

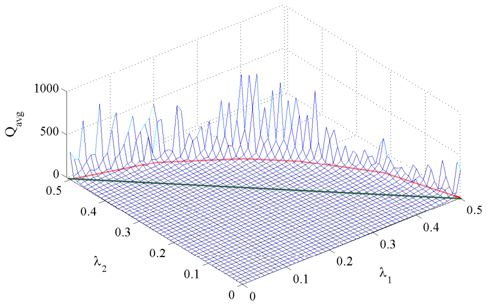

Fig. 8: Average total queue size for the FBDC policy for $p_{10} = p_{01} = 0.20$.

value of 0.9 under the MW policy. Moreover, the FBDC policy with frame length 10 has large queue lengths only for sum-throughputs greater than the maximum sum-throughput value of 1.14 and the throughput loss due to the fixed frame length of 10 is negligible. Fig. 7(b) shows delay as a function of sum-throughput for the FBDC policy with frame length 10 and the MW policy for $p_{10} = p_{01} = 0.10$. While confirming the results of Fig. 7(a), Fig. 7(b) also shows that the stability region becomes larger with increasing channel memory.

Fig. 8 presents the total average queue length, $\mathbf{Q}_{avg} \triangleq \sum_{t=1}^{100K}(Q_1(t) + Q_2(t))/100K$, under the FBDC policy with frame length 25 for $p_{10} = p_{01} = 0.20$ for the simple two-queue system introduced in Section III-A1. The boundary of the stability region is shown by (red) lines on the two dimensional $\lambda_1 - \lambda_2$ plane. We observe that the average queue lengths are small for all $(\lambda_1, \lambda_2) \in \mathbf{\Lambda}$ and the big jumps in queue lengths occur for points outside $\mathbf{\Lambda}$. Finally, the stability region is much larger than the stability region for the corresponding system with i.i.d. channel processes with the same steady state, which is represented by the diagonal line segment between the points $(0, 0.5)$ and $(0.5, 0)$.

V. Conclusions

We investigated the optimal scheduling problem for systems with *reconfiguration delays*, *time-varying channels*, and *interference constraints*. We characterized the stability region of the system in closed form for the case of i.i.d. channel processes and proved that a variable-size frame-based Max-Weight algorithm that makes scheduling decisions based on the queue lengths and the *average channel qualities* is throughput-optimal. For the case of Markovian channels with memory, we characterized the system stability region using state-action frequencies which are stationary solutions to an MDP formulation. We developed the novel FBDC policy based on the state-action frequencies and proved that it is throughput-optimal asymptotically in the frame length.

The state-action frequency approach provides a novel framework for stability region characterization and throughput-optimal policy development for general network control systems, with or without reconfiguration delays. In the future, we intend to pursue Myopic policies with low computation complexity and high throughputs as attractive alternatives to the state-action frequency approach.

References

[1] I. F. Akyildiz and X. Wang, "Wireless mesh networks", *Wiley*, 2009.

[2] E. Altman, P. Konstantopoulos, and Z. Liu, "Stability, monotonicity and invariant quantities in general polling systems," *Queuing Sys.*, vol. 11, pp. 35-57, Mar. 1992.

[3] E. Altman and A. Shwartz, "Markov decision problems and state-action frequencies," *SIAM J. on Control and Opt.*, vol. 29, no 4, pp. 786-809, Jul. 1991.

[4] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar, "Convex analysis and optimization," *Boston: Athena Scientific*, 2003.

[5] L. Blake and M. Long, "Antennas: Fundamentals, design, measurement," *SciTech*, 2009.

[6] D. Bertsimas and J. Tsitsiklis, "Introduction to Linear Optimization", *Athena Scientific*, 1997.

[7] A. Brzezinski and E. Modiano, "Dynamic reconfiguration and routing algorithms for IP-over-WDM networks with stochastic traffic," *IEEE Journal of Lightwave Tech.*, vol. 23, no. 10, pp. 3188-3205, Oct. 2005

[8] G. D. Çelik, L. B. Le, and E. Modiano, "Scheduling in parallel queues with randomly varying connectivity and switchover delay," In *Proc. IEEE INFOCOM'11, Mini Conference*, Apr. 2011.

[9] G. D. Çelik, S. Borst, P. Whiting, and E. Modiano, "Variable frame based Max-Weight algorithms for networks with switchover delay," *Proc. IEEE ISIT'11*, June 2011.

[10] G. D. Çelik and E. Modiano, "Scheduling in Networks with Arbitrary Interference Constraints, Time-Varying Channels, and Reconfiguration Delays," *Technical Report*, LIDS, MIT, July 2011. [Online] Available: http://stuff.mit.edu/people/gcelik/publications.htm

[11] R. G. Gallager, "Discrete stochastic processes," Kluwer, 1996.

[12] L. Georgiadis, M. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," Now Publishers, 2006.

[13] E. N. Gilbert, "Capacity of burst-noise channels," *Bell Syst. Tech. J.*, vol. 39, pp. 12531265, Sep. 1960.

[14] Hawwar et al., "3G UMTS wireless system physical layer: Baseband processing hardware implementation perspective," *IEEE Comm. Magazine - Topics in Radio Comm.*, vol. 44, no. 9, pp. 52-58, Sep. 2006.

[15] S. Mannor and J. N. Tsitsiklis, "On the emperical state-action frequencies in Markov Decision Processes under general policies," *Mathematics of Operation Research*, vol. 30, no. 3, Aug. 2005.

[16] E. Modiano and R. Barry, "A novel medium access control protocol for WDM-based LAN's and access networks using a Master/Slave scheduler," *IEEE J. Lightwave Tech.*, vol. 18, no. 4, pp. 461–468, Apr. 2000.

[17] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," Morgan and Claypool, 2010.

[18] M. J. Neely, E. Modiano, and C. E. Rohrs, "Power allocation and routing in multi-beam satellites with time-varying channels," *IEEE Trans. Netw.*, vol. 11, no. 1, pp. 138–152, Feb. 2003.

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

[20] M. Puterman, "Markov Decision Processes: Discrete stochastic dynamic programming," *Wiley*, 2005.

[21] D. Shah and D. J. Wischik, "Optimal scheduling algorithms for input-queued switches," In *Proc. IEEE INFOCOM'06*, Mar. 2006.

[22] L. Ying, and S. Shakkottai, "On throughput-optimality with delayed network-state information," In *Proc. ITA'08*, Jan. 2008.

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

[24] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Trans. Infor. Theory*, vol. 39, no. 2, pp. 466-478, Mar. 1993.

[25] A. Tolkachev, V. Denisenko, A. Shishlov, and A. Shubov, "High gain antenna systems for millimeter wave radars with combined electronical and mechanical beam steering," In *Proc. IEEE Symp. Phased Array Sys. Tech.*, Oct. 2006.

[26] V. M. Vishnevskii and O. V. Semenova, "Mathematical methods to study the polling systems," *Auto. and Rem. Cont.*, vol. 67, no. 2, pp. 173-220, Feb. 2006.

[27] M. Yun, Y. Zhou, A. Arora, and H. Choi, "Channel-assignment and scheduling in wireless mesh networks considering switching overhead", In *Proc. IEEE ICC'09*, Jun. 2009.