This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2874110, IEEE Journal on Selected Areas in Communications

Delay-Constrained Input-Queued Switch

Lei Deng, *Member, IEEE,* Wing Shing Wong, *Fellow, IEEE,* Po-Ning Chen, *Senior Member, IEEE,* Yunghsiang S. Han, *Fellow, IEEE,* and Hanxu Hou, *Member, IEEE*

***Abstract*—In this paper, we study the delay-constrained input-queued switch where each packet has a deadline and it will expire if it is not delivered before its deadline. Such new scenario is motivated by the proliferation of real-time applications in multimedia communication systems, tactile Internet, networked controlled systems, and cyber-physical systems. The delay-constrained input-queued switch is completely different from the well-understood delay-unconstrained one and thus poses new challenges. We focus on three fundamental problems centering around the performance metric of *timely throughput*: (i) how to characterize the capacity region? (ii) how to design a feasibility/throughput-optimal scheduling policy? and (iii) how to design a network-utility-maximization scheduling policy? We use three different approaches to solve these three fundamental problems. The first approach is based on Markov Decision Process (MDP) theory, which can solve all three problems. However, it suffers from the curse of dimensionality. The second approach breaks the curse of dimensionality by exploiting the combinatorial features of the problem. It gives a new capacity region characterization with only a polynomial number of linear constraints. The third approach is based on the framework of Lyapunov optimization, where we design a polynomial-time maximum-weight T-disjoint-matching scheduling policy which is proved to be feasibility/throughput-optimal. Our three approaches apply to the frame-synchronized traffic pattern but our MDP-based approach can be extended to more general traffic patterns.**

I. Introduction

Switches, which interconnect multiple devices, are the core of communication networks. There are mainly three types of switch designs: output-queued switch, direct input-queued switch, and input-queued switch using virtual output queueing. Among them, the input-queued switch using virtual output queueing is most widely used because it addresses the N-speedup problem of the output-queued switch [2], [3] and the Head-Of-Line (HOL) blocking problem of the direct input-queued switch [4]. In this work, we study the input-queued switch using virtual output queueing, which we simply call *input-queued switch* for the sake of convenience.

Most existing works on input-queued switches consider *delay-unconstrained* traffic where packets can be kept in the virtual output queues forever. Throughput and average delay are two major performance metrics for delay-unconstrained input-queued switches. The authors in [5] characterized the capacity region for independent, identically distributed (i.i.d.) arrivals and further proved that the maximum-weight-matching scheduling policy is *throughput-optimal* in the sense that it can support any feasible throughput requirements in the capacity region. The authors in [6] extended these results to arbitrary delay-unconstrained arrivals by using fluid model techniques. To study the average delay performance, the authors in [7] proposed another throughput-optimal scheduling policy and showed that it attains $O(\log N)$ average delay for $N \times N$ input-queued switches.

However, with the proliferation of real-time application-s, the communication networks nowadays need to support more and more *delay-constrained* traffic. Typical examples include multimedia communication systems such as real-time streaming and video conferencing [8], tactile Internet [9], [10], networked controlled systems (NCSs) such as remote control of unmanned aerial vehicles (UAVs) [11], [12], and cyber-physical systems (CPSs) such as medical tele-operations, X-by-wire vehilces/avionics, factory automation, and robotic collaboration [3]. In such applications, each packet has a hard deadline: if it is not delivered before its deadline, its validity will expire and it will be removed from the system. In addition, throughput, which is termed *timely throughput* in the delay-constrained scenario [13], [14], [15], [8], is also important to such applications. Taking NCSs as an example, the control system can be stabilized if the control messages arrive before the predetermined deadlines and the dropout rate is below a threshold (which equivalently means that the timely throughput is above a threshold) [16], [17]. Taking tactile Internet as another example, the timely throughput is a measure of reliability [10].

Since switches are the core of communication networks, how to support delay-constrained traffic in switches becomes critical. Note that switches can serve delay-constrained traffi c such as tactile applications from both wireless ends and wireline ends. There are some existing works that investigate how to design real-time input-queued switch, e.g., [18], [19], [3]. In [18], the authors proposed two scheduling policies under which the delivery delay of packets is upper bounded by a finite value. In [19], [3], the design goal is to deliver *all* packets and minimize the maximum delivery delay among all packets. Thus, existing works do not directly guarantee the delivery of delay-constrained traffics where hard deadlines are predetermined by the applications; and they do not allow any packet loss. Instead, in this work, we consider how to deliver delay-constrained traffic and focus on the performance metric of timely throughput. More specifically, we study the

This work was partially supported by Schneider Electric, Lenovo Group (China) Limited and the Hong Kong Innovation and Technology Fund (ITS/066/17FP) under the HKUST-MIT Research Alliance Consortium, by the Research Grants Council of the Hong Kong Special Administrative Region under Project GRF 14200217, and by the NSFC of China (No. 61671007 and No. 61701115). A two-page preliminary version of this paper was published as a poster paper in ACM MobiHoc 2018 [1].

L. Deng, Y. S. Han (corresponding author) and H. Hou are with School of Electrical Engineering & Intelligentization, Dongguan University of Technology (email: denglei@dgut.edu.cn, yunghsiangh@gmail.com, hhx@dgut.edu.cn).

W. S. Wong is with Department of Information Engineering, The Chinese University of Hong Kong (email: wswong@ie.cuhk.edu.hk).

P.-N. Chen is with Department of Electrical and Computer Engineering, National Chiao Tung University (email: poning@faculty.nctu.edu.tw).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2874110, IEEE Journal on Selected Areas in Communications

TABLE I
OUR THREE APPROACHES TO SOLVING THE THREE FUNDAMENTAL PROBLEMS FOR DELAY-CONSTRAINED INPUT-QUEUED SWITCHES.

Approach	Capacity Region	Feasibility/Throughput-Optimal Scheduling Policy	Network-Utility-Maxi. Scheduling Policy	Complexity	Extend to General Traffic Pattern
MDP-based (Sec. III)	✓	✓	✓	Exponential	✓
Combinatorial (Sec. IV)	✓	✗	✗	Polynomial	✗
Lyapunov-based (Sec. V)	✗	✓	✗	Polynomial	✗

following three fundamental problems for delay-constrained input-queued switches:

- First, we aim to characterize the capacity region in terms of timely throughput of all input-output pairs. The capacity region serves as the foundation to evaluate the performance of any scheduling policy.
- Second, we aim to design a throughput-optimal (which is termed *feasibility-optimal* in the delay-constrained scenario [13], [14], [15], [8]) scheduling policy, which can support any feasible timely throughput requirements in the capacity region. This problem is important for inelastic applications which have stringent minimum timely throughput requirements.
- Third, we aim to design a scheduling policy to maximize the network utility with respect to the achieved timely throughput. This problem is important for elastic applications which do not have stringent minimum timely throughput requirements but aim to obtain large utility. Here an elastic application has a utility function which increases as its achieved timely throughput increases.

To the best of our knowledge, this is the first presented study on these three fundamental problems centering around timely throughput for delay-constrained input-queued switches. We should emphasize that delay-constrained input-queued switches are completely different from delay-unconstrained ones. In delay-unconstrained scenarios, since packets will never expire and can be kept in the queues forever, the arrival traffic pattern does not make a big difference (actually only the arrival rate matters in the capacity region characterization and in the throughput-optimal scheduling policy design [6]). However, in delay-constrained scenarios, since packets will expire if they are not scheduled before their deadlines, the arrival traffic pattern has a significant impact on timely throughput. Thus, as compared to delay-unconstrained ones, there are new challenges to study delay-constrained input-queued switches.

In this work, as a first step toward answering the above three fundamental problems for delay-constrained input-queued switches, we mainly study a special traffic pattern, called *frame-synchronized traffic pattern*. Such a traffic pattern can find applications in CPSs [20]. It was also the first focus in delay-constrained wireless communication [13], [14], [14], [8]. We also discuss how to consider more general traffic patterns. In this work, we use three different approaches to study the above three fundamental problems. The three approaches come from different angles and all have their own merits. We summarize the results in Table I and detail them as follows:

- The first approach is based on Markov Decision Process (MDP) theory. MDP has a strong modeling capability. Since our system is Markovian (though deterministic), we can use MDP to model our problem. By leveraging results in [8], in Sec. III, we characterize the capacity region, design a feasibility-optimal scheduling policy, and design a network-utility-maximization scheduling policy. Due to its strong modeling capability, the MDP-based approach can be extended to more general traffic pattern, similar to [8]. However, the MDP approach suffers from *the curse of dimensionality*: it has an exponential complexity with the switch size.
- The second approach exploits the problem's combinatorial features. By leveraging some results in combinatorial matrix theory, in Sec. IV, we characterize the capacity region with only a polynomial number of linear constraints (see (6)). This breaks the curse of dimensionality of the first MDP-based approach for capacity region characterization.
- The third approach is based on the framework of Lyapunov optimization. By leveraging the Lyapunov-drift theorem [21], in Sec. V, we show that the problem of minimizing Lyapunov drift is a maximum-weight T-disjoint-matching problem. We further design a polynomial-time algorithm to optimally solve the maximum-weight T-disjoint-matching problem based on the bipartite-graph edge-coloring algorithm. We show that our maximum-weight T-disjoint-matching scheduling policy (called T-MWM) is feasibility-optimal.

We remark that although it is straightforward to apply the MDP-based approach in [8] to solve our three fundamental problems, the solutions are of exponential complexity and thus cannot be efficiently applied to large-size switches. Therefore, the polynomial-time capacity region characterization in (6) and the polynomial-time feasibility-optimal T-MWM scheduling policy are two main contributions of this paper. These two results also serve as the delay-constrained counterparts of the capacity region characterization and the throughput-optimal maximum-weight-matching scheduling policy for the delay-unconstrained input-queued switch in [5].

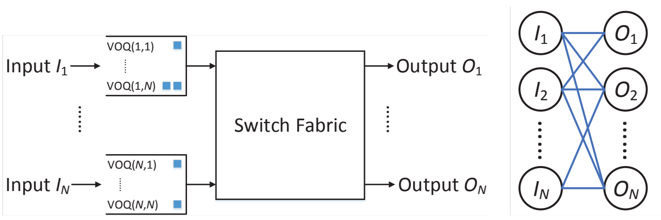
Notation. In this paper, we define set $[C] \triangleq \{1, 2, \cdots, C\}$ for any positive integer C. We use calligraphy font to denote sets, e.g., $\mathcal{A}$. We use bold math font to denote vectors and matrices whose entries use the corresponding normal font, e.g., $\boldsymbol{b} = (b_t : t \in [T]), \boldsymbol{R} = (R_{i,j} : i, j \in [N])$. We sometimes omit the index range of vectors/matrices if it is not ambiguous in the context, e.g., $\boldsymbol{b} = (b_t), \boldsymbol{R} = (R_{i,j})$. We use upper-case letter to denote random variables, e.g., S.

II. SYSTEM MODEL AND PROBLEM FORMULATIONS

A. System Model

Input-Queued Switch. We consider an $N \times N$ input-queued switch using virtual output queueing as shown in Fig. 1(a). Each input I_i has N virtual output queues (VOQs), denoted

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2874110, IEEE Journal on Selected Areas in Communications



(a) Input-queued switch (b) Bipartite graph

Fig. 1. An $N \times N$ input-queued switch using virtual output queueing (VOQ) and its corresponding bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

as VOQ$(i,j), \forall j \in [N]$. VOQ(i,j) contains all packets from input I_i to output O_j.

Traffic Pattern. We consider a time-slotted system. We assume a *frame-synchronized* traffic pattern [13]: starting from slot 1, there is an incoming packet for each VOQ every T slots and the deadline of any packet is also T slots. We call T the frame length. Such a traffic pattern is shown in Fig. 2. If a packet is delivered before its deadline, it contributes to the throughput; otherwise, the packet is useless and will be dropped/discarded from the system.

The frame-synchronized traffic pattern can find applications in CPSs [20]. In addition, like the delay-constrained wireless communication community [13], [14], [15], [8], the special frame-synchronized traffic pattern is a good starting point to investigate delay-constrained input-queued switches. We also show that our first approach (the MDP-based approach) can be extended to more general traffic patterns in Sec. III.

Scheduling Algorithm/Policy. In each slot, the switch fabric can transmit some packets from the inputs to the outputs. In this paper, we use the most common crossbar switch fabric. However, due to the physical limitations of crossbar switch fabric, each input can transmit at most one packet per slot and each output can receive at most one packet per slot. This is also known as the crossbar constraints [7]. The crossbar switch is non-blocking in the sense that all packets satisfying the crossbar constraints can be routed simultaneously in a slot. For the $N \times N$ input-queued switch, we can construct a corresponding bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ between the N inputs and the N outputs where $\mathcal{V} = \{I_1, I_2, \cdots, I_N\} \cup \{O_1, O_2, \cdots, O_N\}$ and $\mathcal{E} = \{(I_i, O_j) : i, j \in [N]\}$, as shown in Fig. 1(b). Then the (deterministic) decision in each slot corresponds to a matching[1] in the bipartite graph $\mathcal{G}$. More specifically, we denote a matching as a matrix $\boldsymbol{M} = (M_{i,j} : i, j \in [N])$, where $M_{i,j} = 1$ if edge (I_i, O_j) is in the matching (i.e., VOQ(i,j) is selected) and $M_{i,j} = 0$ otherwise. Clearly, matching $\boldsymbol{M}$ should satisfy[2]

$$\sum_{j=1}^{N} M_{i,j} \leq 1, \forall i \in [N], \tag{1a}$$

$$\sum_{i=1}^{N} M_{i,j} \leq 1, \forall j \in [N], \tag{1b}$$

$$M_{i,j} \in \{0,1\}, \forall i, j \in [N], \tag{1c}$$

[1]Recall that a matching in a graph is a set of pairwise non-adjacent edges; namely, no two edges share a common vertex.

[2]With a little bit abuse of notation, here we refer matrix $\boldsymbol{M}$ as the edge set in this matching and thus we call it matching $\boldsymbol{M}$.

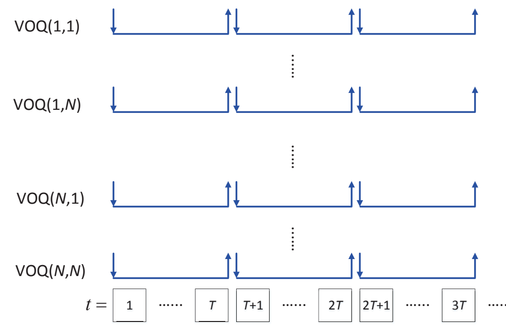


Fig. 2. The frame-synchronized traffic pattern for the input-queued switch.

where (1a) restricts that any input I_i can at most transmit one packet to one output and (1b) restricts that any output O_j can at most receive one packet from one input. We denote the set of all matchings as $\mathcal{M}$, i.e.,

$$\mathcal{M} \triangleq \{\boldsymbol{M} = (M_{i,j} : i, j \in [N]) : \boldsymbol{M} \text{ satisfies (1a)} - \text{(1c)}\}.$$

The decision could also be randomized in that it could randomly choose a matching among multiple matchings. A scheduling algorithm/policy is the set of (possibly randomized) decisions at all slots. We give two definitions for later analysis.

***Definition** 1:* Two matchings $\boldsymbol{M} = (M_{i,j})$ and $\boldsymbol{M}' = (M'_{i,j})$ are *disjoint* if there does not exist a position (i,j) such that both $M_{i,j} = 1$ and $M'_{i,j} = 1$.

***Definition** 2:* If $\boldsymbol{M}^t = (M^t_{i,j})$ is a matching for any $t \in [T]$, we call the collection $\{\boldsymbol{M}^t : t \in [T]\}$ a T*-disjoint matching* if any two of them are disjoint.

B. Problem Formulations

For a scheduling policy π, we define the *timely throughput* [13], [8] from input I_i to output O_j as[3]

$$R^{\pi}_{i,j} \triangleq \liminf_{t \to \infty} \frac{\mathbb{E}\left[\sum_{\tau=1}^{t} D^{\pi}_{i,j,\tau}\right]}{t}, \forall i, j \in [N], \tag{2}$$

where $D^{\pi}_{i,j,\tau} = 1$ if a packet is delivered from input I_i to output O_j at slot τ under scheduling policy π and $D^{\pi}_{i,j,\tau} = 0$ otherwise. Here the expectation is taken over the randomness of matchings if randomized matchings are specified in the scheduling policy π. Since all expired packets will be removed from the system, the timely throughput $R^{\pi}_{i,j}$ is the per-slot average number of delivered packets before expiration for VOQ(i,j). Note that we allow packet dropout/expiration and thus do not need to deliver all traffic packets. However, packet dropout/expiration affects the timely throughput.

A rate matrix $\boldsymbol{R} = (R_{i,j})$ is *feasible* if there exists a scheduling policy such that the timely throughput from input I_i to output O_j is at least $R_{i,j}$ for all $i, j \in [N]$. We then define the *capacity region* $\mathcal{R}(T)$ as the set of all feasible rate matrices with frame length T.

Based on these definitions, in this paper, we study the following three timely-throughput-centric fundamental problems:

- How to characterize the capacity region $\mathcal{R}(T)$?

[3]We also call it the timely throughput of VOQ(i,j).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2874110, IEEE Journal on Selected Areas in Communications

- How to design a feasibility-optimal scheduling policy, i.e., to design a policy that can support any feasible rate matrix $\boldsymbol{R} \in \mathcal{R}(T)$?
- How to design a scheduling policy to maximize the network utility, i.e.,

$$\max_{\boldsymbol{R}\in\mathcal{R}(T)} \sum_{i=1}^{N}\sum_{j=1}^{N} U_{i,j}(R_{i,j}), \tag{3}$$

  where each input-output pair (i,j) has an increasing, concave, and continuously differentiable utility function $U_{i,j}(R_{i,j})$ with respect to its achieved timely throughput $R_{i,j}$?

The capacity region problem is important because it serves as the foundation to evaluate any scheduling policy. The feasibility-optimal scheduling policy design problem is important for inelastic delay-constrained applications which have stringent minimum timely throughput requirements. The network-utility-maximization scheduling policy design problem is important for elastic delay-constrained applications which do not have stringent minimum timely throughput requirements but obtain larger utility for larger timely throughput. Next we propose three different approaches to solve the above three fundamental problems for delay-constrained input-queued switches.

III. An MDP-based Approach

For delay-constrained wireless communication, the authors in [8] proposed a unified MDP-based formulation to study three fundamental problems similar to ours. By observing our system is also Markovian (though deterministic), we can also use MDP theory [22] to solve our three fundamental problems. Since it is similar to apply the MDP-based approach of [8] to our problems, we present the details in our technical report [23], where we show that in principle the MDP-based approach solves all three fundamental problems in Sec. II-B.

In addition, although we mainly study the frame-synchronized traffic pattern, we should further remark that our MDP-based approach can also be extended to more general traffic patterns which might be non-framed or non-synchronized and could have stochastic arrivals. This is similar to [8], which extends the frame-synchronized traffic pattern to general traffic patterns for delay-constrained wireless communication problems.

However, MDP framework suffers from the curse of dimensionality: the number of states is 2^{N^2} and the number of actions is $N!$, both increasing exponentially with respect to the switch size N. Specifically, the MDP-based capacity region characterization has $O(T \cdot 2^{N^2} \cdot N!)$ linear equalities/inequalities and the per-frame time complexity of the MDP-based scheduling policy is $O(N!)$. To break the curse of dimensionality, next in Sec. IV, we exploit the combinatorial features of our problem which are hidden by our MDP-based approach and give a new capacity region characterization with only a polynomial number of linear constraints; and in Sec. V, we propose a polynomial-time feasibility-optimal scheduling policy.

IV. A Simple Capacity Region Characterization

In this section, by exploiting the combinatorial features of the problem, we give a simple capacity region characterization in terms of only a polynomial number of linear constraints for the delay-constrained input-queued switches. Toward that end, we first present some preliminary definitions and results.

***Definition** 3 ([24]):* An $N \times N$ square matrix $\boldsymbol{E} = (E_{i,j})$ is doubly substochastic if it satisfies the following conditions:

$$\begin{cases} \sum_{j=1}^{n} E_{i,j} \le 1, & \forall i \in [N]; \\ \sum_{i=1}^{n} E_{i,j} \le 1, & \forall j \in [N]; \\ E_{i,j} \ge 0, & \forall i,j \in [N]. \end{cases} \tag{4}$$

Denote $\mathcal{J}$ as the set of all doubly substochastic $N \times N$ matrices and let k be a positive integer. Let $\mathcal{J}_k$ be the set of all $1/k$-bounded doubly substochastic $N \times N$ matrices, i.e.,

$$\mathcal{J}_k \triangleq \{\boldsymbol{E} \in \mathcal{J} : E_{i,j} \in [0, 1/k], \forall i,j \in [N]\}. \tag{5}$$

Denote $\mathcal{H}_k$ as the set of all matrices in $\mathcal{J}_k$ whose entries are either 0 or $1/k$. Clearly, $\mathcal{H}_k$ is a finite set. We now give a convex-hull characterization for set $\mathcal{J}_k$.

***Lemma** 1 ([25, Theorem 1]):* $\mathcal{J}_k$ is the convex hull of all matrices in $\mathcal{H}_k$.

Lemma 1 shows that any matrix $\boldsymbol{E} \in \mathcal{J}_k$ can be expressed as a convex combination of some matrices in $\mathcal{H}_k$.

A matrix is a subpermutation matrix if it is a $\{0,1\}$ matrix and each of its line (row or column) has at most one 1. It is straightforward to see that matrix $\boldsymbol{M}$ is a matching (i.e., $\boldsymbol{M} \in \mathcal{M}$) if and only if $\boldsymbol{M}$ is a subpermutation matrix. In addition, a matrix is a k-subpermutation matrix for some positive integer k if it is a $\{0,1\}$ matrix and the sum of each line (row or column) is at most k. We give a decomposition result for k-subpermutation matrices.

***Lemma** 2 ([24, Theorem 4.4.3]):* Any k-subpermutation matrix can be expressed as the sum of k subpermutation matrices.[4]

With the help of the above-mentioned results, we now give a new capacity region characterization for our delay-constrained input-queued switch.

***Theorem** 1:* The capacity region $\mathcal{R}(T)$ is the set of all rate matrices $\boldsymbol{R} = (R_{i,j})$ satisfying the following linear inequalities:

$$\sum_{i=1}^{N} R_{i,j} \le 1, \forall j \in [N], \tag{6a}$$

$$\sum_{j=1}^{N} R_{i,j} \le 1, \forall i \in [N], \tag{6b}$$

$$R_{i,j} \in [0, 1/T], \forall i,j \in [N]. \tag{6c}$$

Proof: The necessity of this result can be easily proved. Since any output O_j can at most receive one packet per slot, the aggregate timely throughput involving output O_j is at most 1 and thus (6a) holds; since any input I_i can at most transmit

[4] In the original result [24, Theorem 4.4.3], the maximum line sum of the matrix is exactly k. However, if the maximum line sum of the matrix is less than k, [24, Theorem 4.4.3] shows that we can decompose it into less than k subpermutation matrices. We can further add some zero matrices such that we can decompose it into exactly k subpermutation matrices.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2874110, IEEE Journal on Selected Areas in Communications

one packet per slot, the aggregate timely throughput involving input I_i is at most 1 and thus (6b) holds; since every VOQ has only one packet in a frame of T slots, its (per-slot) timely throughput is at most $1/T$ and thus (6c) holds. Thus, any feasible rate matrix $\boldsymbol{R}$ must satisfy (6). Then we only need to show that any rate matrix $\boldsymbol{R}$ satisfying (6) can be achieved by some scheduling policy.

Clearly any matrix $\boldsymbol{R}$ satisfying (6) is a $1/T$-bounded doubly substochastic matrix. Then from Lemma 1, we know that $\boldsymbol{R}$ can be expressed as a convex combination of a finite number of (say in total K) doubly substochastic matrices whose entries are either 0 or $1/T$, i.e.,

$$\boldsymbol{R} = \sum_{k=1}^{K} \lambda_k \boldsymbol{R}_k, \tag{7}$$

where $\lambda_k > 0, \sum_{k=1}^{K} \lambda_k = 1$ and matrix $\boldsymbol{R}_k$ is a doubly substochastic matrix with entries being 0 or $1/T$. Since $\boldsymbol{R}_k$ is a doubly substochastic matrix, it has at most T entries being $1/T$ in each line (row or column).

We multiply matrix $\boldsymbol{R}_k$ by T and obtain matrix $T\boldsymbol{R}_k$. Clearly, the entry of matrix $T\boldsymbol{R}_k$ is either 0 or 1 and the sum of each line (row or column) is at most T, implying that $T\boldsymbol{R}_k$ is a T-subpermutation matrix. Now according to Lemma 2, matrix $T\boldsymbol{R}_k$ can be decomposed as the sum of T subpermutation matrices, i.e.,

$$T\boldsymbol{R}_k = \sum_{t=1}^{T} \boldsymbol{M}_{k,t}, \tag{8}$$

where $\boldsymbol{M}_{k,t}$ is a subpermutation matrix, which corresponds to a matching. In addition, since the entry of matrix $T\boldsymbol{R}_k$ is either 0 or 1, all subpermutation matrices (matchings) $\boldsymbol{M}_{k,t}$'s are pairwise disjoint (see Definition 1), implying that $\{\boldsymbol{M}_{k,t} : t \in [T]\}$ is a T-disjoint-matching (see Definition 2).

Combining (7) and (8), we have

$$T\boldsymbol{R} = \sum_{k=1}^{K} \lambda_k \cdot T\boldsymbol{R}_k = \sum_{k=1}^{K} \lambda_k \sum_{t=1}^{T} \boldsymbol{M}_{k,t}. \tag{9}$$

Then we construct the following scheduling policy: in each frame, select the T-disjoint-matching $\{\boldsymbol{M}_{k,t} : t \in [T]\}$ with probability λ_k for any $k \in [K]$. Here when we select the T-disjoint-matching $\{\boldsymbol{M}_{k,t} : t \in [T]\}$ in any frame $f = 0, 1, 2, \cdots$, we do the scheduling as follows:

- Perform matching $\boldsymbol{M}_{k,1}$ at slot $fT + 1$;
- Perform matching $\boldsymbol{M}_{k,2}$ at slot $fT + 2$;
- $\cdots$
- Perform matching $\boldsymbol{M}_{k,T}$ at slot $(f + 1)T$.

For the T-disjoint-matching $\{\boldsymbol{M}_{k,t} : t \in [T]\}$, if

$$\left(\sum_{t=1}^{T} \boldsymbol{M}_{k,t}\right)_{i,j} = (T\boldsymbol{R}_k)_{i,j} = 1,$$

VOQ(i, j) will be scheduled in a frame; otherwise, VOQ(i, j) will not be scheduled. Since we select all (in total K) T-disjoint-matchings randomly according to probability distribution $\{\lambda_k\}$, the probability to schedule VOQ(i, j) (which is also the expected number of delivered packets for VOQ(i, j)) in a frame is

$$\sum_{k=1}^{K} \lambda_k (T\boldsymbol{R}_k)_{i,j} = (T\boldsymbol{R})_{i,j} = TR_{i,j}, \tag{10}$$

where the first equality follows from (9). Therefore, the (per-slot) timely throughput of VOQ(i, j) is $\frac{TR_{i,j}}{T} = R_{i,j}$ for any VOQ(i, j). This completes the proof. ■

Theorem 1 gives a new capacity region characterization (6) with only $2N^2+2N = O(N^2)$ linear inequalities, much lower than the exponential-size MDP-based characterization (which needs $O(T \cdot 2^{N^2} \cdot N!)$ linear equalities/inequalities). We further make some remarks for Theorem 1.

A. Comparison with Delay-Unconstrained Results

Our capacity region characterization for delay-constrained input-queued switches has a similar non-overbooking condition (see (6a), (6b)) with that for delay-unconstrained ones [5], [6], [26], [27], [28], except that each VOQ's timely throughput is upper bounded by $1/T$ (see (6c)). However, there is a fundamental difference — in delay-unconstrained input-queued switches, the capacity region is in terms of the (incoming) arrival rate of all VOQs, while in our delay-constrained ones, the capacity region is in terms of the (achieved) timely throughput of VOQs. In other words, we allow packet loss/expiration and characterize the fundamental limit of timely throughput in our delay-constrained input-queued switches. We also compare our proof technique for Theorem 1 with that for the capacity region characterization for delay-unconstrained input-queued switches [26], [27] in our technical report [23],

B. The Special Case of $T \geq N$

If $T \geq N$, we can see that the rate matrix $\boldsymbol{R} = (R_{i,j} = 1/T : i, j \in [N])$ is in the capacity region (6), which achieves the largest timely throughput for all VOQs. In fact, (6c) implies (6a) and (6b) when $T \geq N$. Indeed, when $T \geq N$, we can construct a scheduling policy to transmit all packets without any packet loss/expiration so as to attain a timely throughput of $1/T$ for all VOQs. Please see the details of how to construct the scheduling policy in our technical report [23].

C. Lack of Scheduling Policy

Note that when we prove the achievability part in Theorem 1, we construct a randomized scheduling policy based on the distribution $\{\lambda_k\}$. Although we show the existence of parameters $\{\lambda_k\}$, we do *not* know how to find such $\{\lambda_k\}$. Thus, our constructed randomized scheduling policy is only an existing policy but we do not have ways to implement it.

This is different from the result in delay-unconstrained Birkhoff-von Neumann input-queued switches in [26], [27], where the authors utilized the fact that any $N \times N$ doubly stochastic matrix can be expressed as the convex combination of some $N \times N$ permutation matrices [29], and more importantly they proposed an algorithm of complexity $O(N^{4.5})$ to find the convex-combination parameters $\{\phi_k\}$. Based on $\{\phi_k\}$, the authors in [26], [27] further implemented a throughput-optimal scheduling policy in polynomial time.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2874110, IEEE Journal on Selected Areas in Communications

V. A Polynomial-Time Feasibility-Optimal Scheduling Policy

The combinatorial approach in Sec. IV breaks the curse of dimensionality of the MDP-based approach for the problem of characterizing the capacity region. In this section, we further break the curse of dimensionality of the MDP-based approach for the problem of designing a feasibility-optimal scheduling policy. In particular, we leverage the framework of Lyapunov optimization and design a polynomial-time feasibility-optimal scheduling policy for our delay-constrained input-queued switches.

For any VOQ(i,j), if it has a timely throughput requirement $R_{i,j}$, we construct a virtual queue[5] as shown in Fig. 3:

- The virtual queue is indexed by the frames in the real system, denoted as $f=0,1,2,\cdots$;
- The arrival process of the virtual queue $A_{i,j}(f)$ is a constant flow with size $TR_{i,j}$ for any frame f;
- The service process of the virtual queue $B_{i,j}(f)$ depends on the scheduling policy in the real system: $B_{i,j}(f)=1$ if VOQ(i,j) is scheduled in frame f in the real system and $B_{i,j}(f)=0$ otherwise;
- By using the standard queue dynamics in [21], the queue is updated as (with initial queue length $Q_{i,j}(0)=0$)

$$Q_{i,j}(f+1)=\max\{Q_{i,j}(f)-B_{i,j}(f),0\}+A_{i,j}(f). \quad (11)$$

Note that the virtual (queue) system is different from the real system. In the real system, a packet expires at the end of its frame. However, in the virtual queue, all arrivals will not expire and always stay in the virtual queue. Moreover, the time scale is also different: our virtual system is frame-based while our real system is slot-based. We use $B_{i,j}(f)$ to connect the virtual system and real system.

According to the queue stability theorem [21, Theorem 2.5(b)], if the virtual queue $Q_{i,j}$ is mean rate stable, then

$$\limsup_{F\to\infty}\frac{1}{F}\sum_{f=0}^{F-1}\mathbb{E}[A_{i,j}(f)-B_{i,j}(f)]\le 0. \quad (12)$$

Since $A_{i,j}(f)=TR_{i,j},\forall f$, then (12) implies,

$$\liminf_{F\to\infty}\frac{1}{F}\sum_{f=0}^{F-1}\mathbb{E}[B_{i,j}(f)]\ge TR_{i,j}. \quad (13)$$

Note that $\liminf_{F\to\infty}(1/F)\sum_{f=0}^{F-1}\mathbb{E}[B_{i,j}(f)]$ is the achieved per-frame timely throughput for VOQ(i,j) in the real system. Hence the achieved (per-slot) timely throughput for VOQ(i,j) in the real system is

$$\liminf_{F\to\infty}\frac{1}{TF}\sum_{f=0}^{F-1}\mathbb{E}[B_{i,j}(f)]\ge\frac{TR_{i,j}}{T}=R_{i,j}.$$

Thus, to achieve timely throughput $R_{i,j}$ for VOQ(i,j) is equivalent to make the virtual queue $Q_{i,j}$ mean rate stable.

By using the Lyapunov-drift theorem [21, Theorem 4.1], it is standard to show that the following maximum-weight scheduling policy can make all virtual queues mean rate stable:

[5]Readers should distinguish virtual queue here from VOQ (virtual output queue).

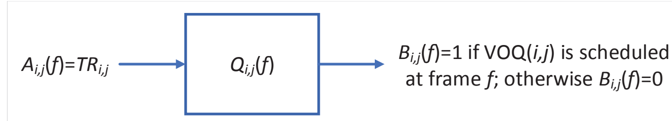


Fig. 3. The constructed virtual queue.

in each frame $f=0,1,2,\cdots$, select a matrix $\boldsymbol{B}(f)=(B_{i,j}(f):i,j\in[N])$ (which corresponds to T matchings in this frame of in total T slots) to maximize the queue weight sum, i.e.,

$$\max_{\boldsymbol{B}(f)}\sum_{i=1}^{N}\sum_{j=1}^{N}Q_{i,j}(f)B_{i,j}(f). \quad (14)$$

Our frame-based maximum-weight problem (14) is different from the slot-based one for delay-unconstrained input-queued switches [5], [6], which is exactly the classical maximum-weight-matching problem. In (14), we need to find T matchings to solve the frame-based maximum-weight problem. Recall that $B_{i,j}(f)=1$ if VOQ(i,j) is selected in frame f. Even if VOQ(i,j) is scheduled more than once in frame f, $B_{i,j}(f)$ is still 1 and cannot increase the objective value in (14). This implies that there is no need to schedule the same VOQ for more than once in a frame. Thus, it suffices to find T *disjoint* matchings to solve (14), i.e., to select $\boldsymbol{B}(f)$ in frame f is equivalent to select a T-disjoint-matching (see Definition 2) of the bipartite graph $\mathcal{G}$.

Since in each frame we need to solve the same problem (14) (though with different queue lengths/weights), let us ignore the frame index f. The problem to find a T-disjoint-matching with virtual queue weights $(Q_{i,j})$ to maximize the queue weight sum can be formulated as an integer linear programming (ILP),

$$\max \quad \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N}Q_{i,j}b_{i,j}^{t} \quad (15a)$$

$$\text{s.t.} \quad \sum_{i=1}^{N}b_{i,j}^{t}\le 1,\forall j\in[N],t\in[T] \quad (15b)$$

$$\sum_{j=1}^{N}b_{i,j}^{t}\le 1,\forall i\in[N],t\in[T] \quad (15c)$$

$$\sum_{t=1}^{T}b_{i,j}^{t}\le 1,\forall i,j\in[N] \quad (15d)$$

$$\text{var.} \quad b_{i,j}^{t}\in\{0,1\},\forall i,j\in[N],t\in[T] \quad (15e)$$

In (15), constraints (15b) and (15c) restrict that at any slot $t\in[T]$, we select a matching $\boldsymbol{b}^t=(b_{i,j}^t)\in\mathcal{M}$; constraint (15d) restricts that all matchings selected in T slots of the frame are pairwise disjoint, i.e., $\{\boldsymbol{b}^t:t\in[T]\}$ is a T-disjoint-matching. Based on (15), we can simply reconstruct $B_{i,j}=\sum_{t=1}^{T}b_{i,j}^t$ to solve problem (14).

A nature approach to solve ILP (15) is to iteratively apply the (per-slot) maximum-weight matching algorithm. However, as we show in our technical report [23], the greedy iterative maximum-weight-matching algorithm is strictly suboptimal to ILP (15). This indicates that it is nontrivial to solve ILP (15). To solve ILP (15) optimally and efficiently, we establish

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2874110, IEEE Journal on Selected Areas in Communications

equivalence between ILP (15) and the following new ILP:

$$\max \quad \sum_{i=1}^{N}\sum_{j=1}^{N} Q_{i,j}c_{i,j} \tag{16a}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} c_{i,j} \leq T, \forall j \in [N] \tag{16b}$$

$$\sum_{j=1}^{N} c_{i,j} \leq T, \forall i \in [N] \tag{16c}$$

$$\text{var.} \quad c_{i,j} \in \{0,1\}, \forall i,j \in [N] \tag{16d}$$

In (16), we find a set of VOQs to maximize the sum of their queue length/weight such that each input/outout is incident to at most T VOQs. From the perspective of bipartite graph $\mathcal{G}$, ILP (16) is to find a set of edges to maximize their weight sum such that each node is incident to at most T edges. We now establish the equivalence between ILP (15) and ILP (16).

***Theorem** 2:* The optimal values of ILPs (15) and (16) are the same. Moreover, for any optimal solution $\{c_{i,j}\}$ to (16), we can use the bipartite-graph edge-coloring algorithm to construct an optimal solution $\{b_{i,j}^t\}$ to (15) in polynomial time.

Proof: (i) For any feasible solution $\{b_{i,j}^t\}$ to ILP (15), we construct

$$c_{i,j} = \sum_{t=1}^{T} b_{i,j}^t.$$

We can easily check that $\{c_{i,j}\}$ is feasible to ILP (16) and the objective value of ILP (16) is equal to that of ILP (15) since

$$\sum_{i=1}^{N}\sum_{j=1}^{N} Q_{i,j}c_{i,j} = \sum_{i=1}^{N}\sum_{j=1}^{N} Q_{i,j}\sum_{t=1}^{T} b_{i,j}^t = \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N} Q_{i,j}b_{i,j}^t.$$

Therefore, the optimal value of ILP (16) is an upper bound of the optimal value of ILP (15).

(ii) For any feasible solution $\{c_{i,j}\}$ to (16), we construct a bipartite graph $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ where $(I_i, O_j) \in \mathcal{E}'$ if $c_{i,j} = 1$. Due to (16b) and (16c), we know that the maximum degree of all nodes in $\mathcal{G}'$ is at most T. The edge-coloring problem for a graph is to use minimum number of colors to color all edges such that any two edges sharing a common node do not have the same color. It is well-known that the edges of any bipartite graph can be colored with Δ colors [30], [31], [32] where Δ is the maximum node degree. Thus, our graph $\mathcal{G}'$ can be colored with at most T colors. Clearly, the set of all edges sharing the same color forms a matching of bipartite graph $\mathcal{G}'$ and all such (at most T) matchings are disjoint. For any matching, we can represent it as $(b_{i,j} : i,j \in [N])$ where $b_{i,j} = 1$ if edge (I_i, O_j) is in the matching. We can also add some dummy/empty matchings such that we have in total T disjoint matchings, i.e., constructing a feasible solution $\{b_{i,j}^t : i,j \in [N], t \in [T]\}$ to problem (15). Since all edges in graph $\mathcal{G}'$ is colored by those (at most T) colors, we thus have

$$\sum_{t=1}^{T} b_{i,j}^t = c_{i,j}, \forall i,j \in [N]$$

and further

$$\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N} Q_{i,j}b_{i,j}^t = \sum_{i=1}^{N}\sum_{j=1}^{N} Q_{i,j}\sum_{t=1}^{T} b_{i,j}^t = \sum_{i=1}^{N}\sum_{j=1}^{N} Q_{i,j}c_{i,j}$$

implying that the objective value of ILP (15) is equal to that of ILP (16). Thus the optimal value of ILP (16) is a lower bound of the optimal value of ILP (15).

Part (i) and part (ii) show that the optimal values of ILP (15) and ILP (16) are the same. Thus, the construction in part (ii) for any optimal solution to (16) results in an optimal solution to (15).

It is well-known that we can color the bipartite graph $\mathcal{G}'$ with minimal number of (at most T) colors in polynomial time [30], [31], [32]. The best algorithm is that in [32] with complexity $O(N^2 \log T) = O(N^2 \log N)$.[6] Therefore, once we obtain an optimal solution $\{c_{i,j}\}$ to (16), we can use the bipartite-graph edge-coloring algorithm to construct an optimal solution $\{b_{i,j}^t\}$ to (15) in polynomial time. ■

Theorem 2 shows that we only need to get an optimal solution to (16) in order to get an optimal solution $\{b_{i,j}^t\}$ to (15). Then the remaining problem is whether we can solve the new ILP (16) efficiently. Indeed, problem (16) can be solved in polynomial time.

It was shown in [33] that the constraint matrix of the vectorized version of (16) is totally unimodular. Thus, we

Algorithm 1 The Maximum Weight T-Disjoint-Matching Algorithm (T-MWM)

```
Require: A timely throughput matrix R = (R_{i,j}) ∈ R(T)
for i, j = 1, 2, ··· , N do
    Set Q_{i,j}(0) = 0
end for
for f = 0, 1, 2, ··· do
    Solve the relaxed LP of the vectorized version of ILP
    (16) with weights {Q_{i,j}(f) : i, j ∈ [N]} by the simplex
    algorithm and get a vertex optimal solution {c_{i,j}}
    Construct a bipartite graph G' = (V, E') where
    (I_i, O_j) ∈ E' if c_{i,j} = 1
    Use the bipartite-graph edge-coloring algorithm in
    [32] to color G' and get a T-disjoint-matching
    (possibly inserting some dummy/empty matchings)
    {(b^1_{i,j}), (b^2_{i,j}), ··· , (b^T_{i,j})}
    for t = 1, 2, ··· , T do
        Perform matching (b^t_{i,j}) at slot fT + t
    end for
    for i = 1, 2, ··· , N do
        for j = 1, 2, ··· , N do
            Set B_{i,j}(f) = sum_{t=1}^{T} b^t_{i,j}
            Set Q_{i,j}(f + 1) = max{Q_{i,j}(f) − B_{i,j}(f), 0} +
            T R_{i,j}
        end for
    end for
end for
```

[6]Here we require $T < N$. Note that from the remark given in Sec. IV-B, we know that we can deliver all packets with a simple policy when $T \geq N$. Thus, in this section, we only need to consider $T < N$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2874110, IEEE Journal on Selected Areas in Communications

can resort to solving the relaxed LP of (16) and any vertex optimal solution of the relaxed LP would be integral and thus optimal to (16). For example, the most widely-used simplex algorithm for LP outputs a vertex optimal solution. Moreover, a recent result by Kitahara and Mizuno in [34] shows that for an LP whose constraint matrix is totally unimodular and constraint constant vector is integral (which is indeed our case), the number of different vertex solutions generated by the simplex method for this LP is polynomially bounded by $n\lceil m||b||_1 \log(m||b||_1)\rceil$ where n is the number of variables, m is the number of constraints and b is the constraint constant vector. It is easy to see that $n = N^2$, $m = 2N^2 + 2N$, and $||b||_1 = 2NT + N^2 \leq 3N^2$ for our relaxed LP of (16). Thus, we can solve ILP (16) in polynomial time with complexity $n\lceil m||b||_1 \log(m||b||_1)\rceil \leq N^2\lceil (2N^2+2N)\cdot 3N^2 \cdot \log((2N^2 + 2N)\cdot 3N^2)\rceil = O(N^6 \log N)$.

Readers may wonder whether we can directly solve the relaxed LP of (15), instead of leveraging an intermediate ILP (16). It turns out that the direct approach does not work. We use an example in our technical report [23] to show that the constraint matrix of the vectorized version of our original ILP (15) is *not* totally unimodular. Therefore, establishing the equivalence between ILP (15) and ILP (16) is crucial.

We summarize the proposed maximum-weight T-disjoint-matching scheduling algorithm in Algorithm 1, which we call T-MWM. We now give a theorem to show that our T-MWM Algorithm is feasibility-optimal, i.e., it can achieve any feasible timely throughput requirements $\boldsymbol{R} \in \mathcal{R}(T)$.

***Corollary** 1:* T-MWM is feasibility-optimal.

Proof: Based on Theorem 2 and [33], we conclude that lines 5-7 of Algorithm 1 (T-MWM) solves the maximum-weight scheduling problem (14). Thus, all virtual queues $(Q_{i,j})$ are mean rate stable according to the Lyapunov-drift theorem [21, Theorem 4.1]. Therefore, T-MWM is feasibility-optimal. ∎

Note that the per-frame time complexity of Algorithm 1 (T-MWM) is polynomial in the order of $O(N^6 \log N) + O(N^2 \log N) = O(N^6 \log N)$, which is much faster than the exponential-time MDP-based policy (of order $O(N!)$).

Remarks. In this section, we adopt the Lyapunov-optimization framework to design a polynomial-time feasibility-optimal scheduling policy. We should further remark that our virtual queue $V_{i,j}(f)$ defined in (11) is also termed deficit in the delay-constrained wireless communication community [13], [8], [15]. In particular, our maximum weight scheduling policy is similar to the largest-deficit-first (LDF) scheduling policy. However, as compared with LDF scheduling policy which only needs to select the flow with largest deficit in each slot, our maximum weight scheduling policy needs to solve a more difficult combinatorial problem, i.e., ILP (15). In addition to the capacity region characterization in Theorem 1, our main contribution in this section is to show that ILP (15) is equivalent to another problem, i.e., ILP (16), which can be solved in polynomial time.

VI. Simulation

In this section, we use simulation to evaluate our capacity region and scheduling policies.

First, we show that the capacity region characterized by the MDP-based approach (see the details in our technical report [23]) and the capacity region in (6) characterized by the combinatorial approach are the same. We simulate a 3×3 switch and vary the frame length T from 1 to 5. Since it is difficult to visualize the capacity region (of dimension $3 \times 3 = 9$), we solve the network-utility-maximization problem (3) for two different capacity region characterizations. We adopt a linear utility function $U_{i,j}(R_{i,j}) = w_{i,j} R_{i,j}$ for each VOQ(i,j). We randomly pick a weight matrix, which is realized as

$$\boldsymbol{w} = (w_{i,j}) = \begin{pmatrix} 0.70 & 0.84 & 0.54 \\ 0.51 & 0.92 & 0.44 \\ 0.10 & 0.30 & 0.28 \end{pmatrix}.$$

Note that both the MDP-based approach and the combinatorial approach characterize the capacity region in terms of some linear constraints. Thus, under the linear utility functions, the network-utility maximization problem (3) becomes a linear programming (LP), whose constraints are different under two different capacity region characterizations.

We show the achieved maximum network utility in Fig. 4(a). We can see that under two different capacity region characterizations, the achieved maximum network utilities are the same. Namely, the two LPs with different linear constraints give the same optimal value. We remark that such result holds for all our randomly generated weighted matrices, verifying that our two different capacity region characterizations are the same. In addition, since each VOQ has only 1 packet every T slots, the timely throughput of any VOQ is upper bounded by $1/T$ and we thus plot the utility upper bound $\sum_{i=1}^N \sum_{j=1}^N w_{i,j}/T$ in Fig. 4(a). We can see that indeed when $T \geq N = 3$, the achieved maximum network utility attains the upper bound, verifying our discussion in Sec. IV-B.

Second, we compare our proposed two feasibility-optimal scheduling policies: the MDP-based algorithm (called RCS algorithm, see the details in our technical report [23]) and T-MWM algorithm (Algorithm 1). We again consider a 3×3 switch with $T = 2$ and input a feasible rate matrix,

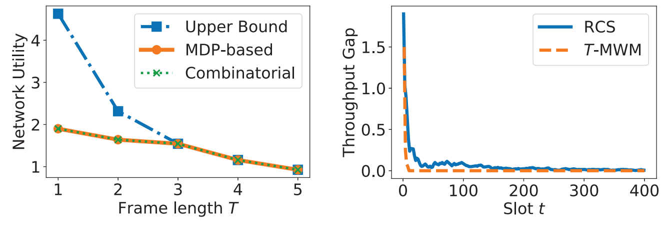
$$\boldsymbol{R} = (R_{i,j}) = \begin{pmatrix} 0.2 & 0.4 & 0.4 \\ 0.3 & 0.5 & 0.2 \\ 0.5 & 0.1 & 0.4 \end{pmatrix}.$$

We then run RCS and T-MWM. To verify that they are feasibility-optimal, we need to show that both can achieve the target rate matrix $\boldsymbol{R}$. For any VOQ(i,j), we obtain the empirical timely throughput up to slot t as

$$R_{i,j}^{\mathsf{emp}}(t) \triangleq \frac{\sum_{\tau=1}^{t} D_{i,j,\tau}}{t},$$

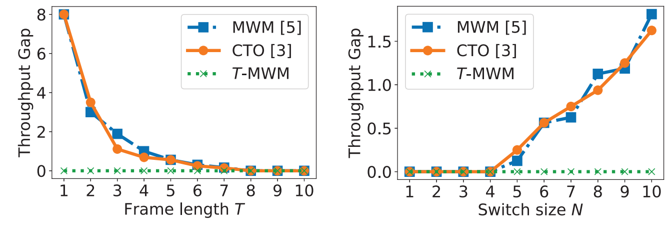
where $D_{i,j,\tau} = 1$ if a packet is delivered from input I_i to output O_j at slot τ and $D_{i,j,\tau} = 0$ otherwise. We thus define the *throughput gap* between the empirical rate matrix $\boldsymbol{R}^{\mathsf{emp}}(t)$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2874110, IEEE Journal on Selected Areas in Communications



(a) (b)

Fig. 4. Simulation for a 3×3 input-queued switch. (a)Verification of the equivalence of two capacity region characterizations; (b) Evaluation for two feasibility-optimal scheduling policies.



(a) (b)

Fig. 5. Compare our proposed T-MWM policy with two baselines: (one-slot) maximum-weight-matching (MWM) scheduling policy that was proposed by [5] for delay-unconstrained input-queued switch and the clearance-time-optimal (CTO) scheduling policy that was proposed by [3] for real-time input-queued switch. (a) The switch size is $N = 8$; (b) The frame length is $T = 4$.

and the target rate matrix $\boldsymbol{R}$ as

$$\delta(\boldsymbol{R}^{\mathsf{emp}}(t), \boldsymbol{R}) \triangleq \sum_{i=1}^{N}\sum_{j=1}^{N} \max\left\{R_{i,j} - R_{i,j}^{\mathsf{emp}}(t), 0\right\}. \quad (17)$$

Clearly, $\delta(\boldsymbol{R}^{\mathsf{emp}}(t), \boldsymbol{R}) > 0$ if and only if there exists a VOQ which does not achieve its target timely throughput, i.e., $\exists i, j \in [N]$ such that $R_{i,j}^{\mathsf{emp}}(t) < R_{i,j}$; and $\delta(\boldsymbol{R}^{\mathsf{emp}}(t), \boldsymbol{R}) = 0$ if and only if every VOQ achieves the target timely throughput, i.e., $R_{i,j}^{\mathsf{emp}}(t) \geq R_{i,j}, \forall i, j \in [N]$. We show the throughput gap for all slots in Fig. 4(b). We can see that the throughput gap converges to 0 in both algorithms, implying that both algorithms achieve the target rate matrix $\boldsymbol{R}$. We remark that such result holds for all our tried feasible rate matrices, verifying that both algorithms are feasibility-optimal.

Finally, we compare our proposed T-MWM scheduling policy with two baselines for the input-queued switch. The first one is the (one-slot) maximum-weight-matching (MWM) scheduling policy that was proposed by [5] for delay-unconstrained input-queued switch. MWM was proved to be throughput-optimal for delay-unconstrained traffic in [5], [6]. The second one is the clearance-time-optimal (CTO) scheduling policy that was proposed by [3] for real-time input-queued switch. The authors in [3] proved that CTO can minimize the maximum delivery delay among all packets (i.e., the clearance time). Note that both MWM and CTO scheduling polices are not designed to route delay-constrained traffic where the hard deadline is specified by different applications. Both MWM and CTO determine the schedule according to the length of real VOQs, while our T-MWM determines the schedule according to the length of virtual queues (11).

To compare these three scheduling policies in the delay-constrained setting, for switch size N and frame length T, we randomly select a weight matrix $\boldsymbol{w}$ and solve the network-utility-maximization problem $\max_{\boldsymbol{R} \in \mathcal{R}} \sum_{i,j \in [n]} w_{i,j} R_{i,j}$, which gives us a feasible rate matrix $\boldsymbol{R}$. We then apply MWM, CTO, and T-MWM scheduling policies to obtain the empirical timely throughput up to 10000 slots and finally we obtain the throughput gap based on (17). We show the throughput gap of the three policies in Fig. 5, where we fix the switch size to be $N = 8$ and vary the frame length T from 1 to 10 in Fig. 5(a) and we fix the frame length to be $T = 4$ and vary the switch size N from 1 to 10 in Fig. 5(b). As we can see, our proposed T-MWM can achieve the target rate matrix $\boldsymbol{R}$ in any case, but neither MWM nor CTO can achieve it when $N > T$. Thus, our proposed T-MWM policy outperforms both baselines when the input-queued switch is required to deliver delay-constrained traffic.

VII. Conclusion

To support delay-constrained traffic of real-time applications such as tactile Internet, networked control systems, and cyber-physical systems, we study how to re-design the input-queued switch, which is the core component of communication networks. We use three different approaches to solve the three fundamental problems for delay-constrained input-queued switches centering around the performance metric of timely throughput. The MDP-based approach can solve all three problems. In addition, the MDP-based approach can also be extended to more general traffic patterns. However, the MDP-based approach suffers from the curse of dimensionality. To address this issue, we propose a combinatorial approach to characterize the capacity region with only a polynomial number of linear constraints and further propose a Lyapunov-based approach to design a polynomial-time feasibility-optimal scheduling policy. In the future, it is important to study how to design a polynomial-time network-utility-maximization scheduling policy, how to efficiently extend to general traffic patterns to capture more practical scenarios, and how to implement our algorithms in practical switches. In addition, it is interesting to study the system behaviour when we apply our algorithms to the real communication system which deliver real-world delay-constrained traffic.

References

[1] L. Deng, W. S. Wong, P.-N. Chen, and Y. S. Han, "Delay-constrained input-queued switch," in *Proc. ACM MobiHoc (Poster Paper)*, 2018.

[2] S.-T. Chuang, A. Goel, N. McKeown, and B. Prabhakar, "Matching output queueing with a combined input/output-queued switch," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 6, pp. 1030–1039, 1999.

[3] K. Kang, K.-J. Park, L. Sha, and Q. Wang, "Design of a crossbar VOQ real-time switch with clock-driven scheduling for a guaranteed delay bound," *Real-Time Systems*, vol. 49, no. 1, pp. 117–135, 2013.

[4] M. Karol, M. Hluchyj, and S. Morgan, "Input versus output queueing on a space-division packet switch," *IEEE Transactions on Communications*, vol. 35, no. 12, pp. 1347–1356, 1987.

[5] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Transactions on Communications*, vol. 47, no. 8, pp. 1260–1267, 1999.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2018.2874110, IEEE Journal on Selected Areas in Communications

[6] J. G. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," in *Proc. IEEE INFOCOM*, 2000.

[7] M. J. Neely, E. Modiano, and Y.-S. Cheng, "Logarithmic delay for $N \times N$ packet switches under the crossbar constraint," *IEEE/ACM Transactions on Networking*, vol. 15, no. 3, pp. 657–668, 2007.

[8] L. Deng, C.-C. Wang, M. Chen, and S. Zhao, "Timely wireless flows with general traffic patterns: Capacity region and scheduling algorithms," *IEEE/ACM Transactions on Networking*, vol. 25, no. 6, pp. 3473–3486, 2017.

[9] G. P. Fettweis, "The tactile Internet: Applications and challenges," *IEEE Vehicular Technology Magazine*, vol. 9, no. 1, pp. 64–70, 2014.

[10] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5G-enabled tactile Internet," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 460–473, 2016.

[11] J. Baillieul and P. J. Antsaklis, "Control and communication challenges in networked real-time systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 9–28, 2007.

[12] Z. Jiang, H. Cheng, Z. Zheng, X. Zhang, X. Nie, W. Li, Y. Zou, and W. S. Wong, "Autonomous formation flight of uavs: Control algorithms and field experiments," in *Proc. CCC*, 2016.

[13] I.-H. Hou, V. Borkar, and P. R. Kumar, "A theory of QoS for wireless," in *Proc. IEEE INFOCOM*, 2009.

[14] I.-H. Hou and P. R. Kumar, "Utility maximization for delay constrained QoS in wireless," in *Proc. IEEE INFOCOM*, 2010.

[15] X. Kang, W. Wang, J. J. Jaramillo, and L. Ying, "On the performance of largest-deficit-first for scheduling real-time traffic in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 72–84, 2016.

[16] C. Tan and H. Zhang, "Necessary and sufficient stabilizing conditions for networked control systems with simultaneous transmission delay and packet dropout," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 4011–4016, 2017.

[17] C. Tan, H. Zhang, and W. S. Wong, "Delay-dependent algebraic riccati equation to stabilization of networked control systems: Continuous-time case," *IEEE Transactions on Cybernetics*, vol. PP, no. 99, pp. 1–12, 2017.

[18] C.-S. Chang, D.-S. Lee, and C.-Y. Yue, "Providing guaranteed rate services in the load balanced Birkhoff-von Neumann switches," *IEEE/ACM Transactions on Networking*, vol. 14, no. 3, pp. 644–656, 2006.

[19] Q. Wang, S. Gopalakrishnan, X. Liu, and L. Sha, "A switch design for real-time industrial networks," in *Proc. IEEE RTAS*, 2008.

[20] K.-D. Kim and P. R. Kumar, "Cyber–physical systems: A perspective at the centennial," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1287–1308, 2012.

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

[22] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[23] L. Deng, W. S. Wong, P.-N. Chen, Y. S. Han, and H. Hou, "Delay-constrained input-queued switch," https://arxiv.org/abs/1809.02826, 2018.

[24] R. A. Brualdi and H. J. Ryser, *Combinatorial Matrix Theory*. Cambridge University Press, 1991.

[25] L. Deng and Q. Lin, "Convex sets of doubly substochastic matrices," https://arxiv.org/abs/1711.06818, 2017.

[26] C.-S. Chang, W.-J. Chen, and H.-Y. Huang, "On service guarantees for input-buffered crossbar switches: A capacity decomposition approach by birkhoff and von neumann," in *Proc. IEEE IWQoS*, 1999.

[27] ——, "Birkhoff-von neumann input buffered crossbar switches," in *Proc. IEEE INFOCOM*, 2000.

[28] C.-S. Chang, D.-S. Lee, and Y.-S. Jou, "Load balanced Birkhoff–von Neumann switches, part i: One-stage buffering," *Computer Communications*, vol. 25, no. 6, pp. 611–622, 2002.

[29] G. Birkhoff, "Tres observaciones sobre el algebra lineal," *Universidad Nacional de Tucumán Revista, Serie A*, vol. 5, pp. 147–151, 1946.

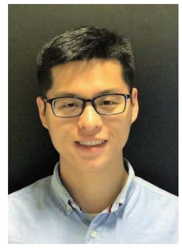
[30] D. König, "Über graphen und ihre anwendung auf determinantentheorie und mengenlehre," *Mathematische Annalen*, vol. 77, no. 4, pp. 453–465, 1916.

[31] A. Schrijver, "Bipartite edge coloring in $O(\Delta m)$ time," *SIAM Journal on Computing*, vol. 28, no. 3, pp. 841–846, 1998.

[32] R. Cole, K. Ost, and S. Schirra, "Edge-coloring bipartite multigraphs in $O(E \log D)$ time," *Combinatorica*, vol. 21, no. 1, pp. 5–12, 2001.

[33] C. J. Taylor, "On the optimal assignment of conference papers to reviewers," 2008.

[34] T. Kitahara and S. Mizuno, "A bound for the number of different basic solutions generated by the simplex method," *Mathematical Programming*, vol. 137, no. 1–2, pp. 579–586, 2013.



Lei Deng (M'17) received the B.Eng. degree from the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2012, and the Ph.D. degree from the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, in 2017. In 2015, he was a Visiting Scholar with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. He is now an assistant professor in School of Electrical Engineering & Intelligentization, Dongguan University of Technology. His research interests are timely network communications, intelligent transportation system, and spectral-energy efficiency in wireless networks.

Wing Shing Wong (M'81–SM'90–F'02) received a combined master and bachelor degree from Yale University and M.S. and Ph.D. degrees from Harvard University. He worked for the AT&T Bell Laboratories from 1982 until he joined the Chinese University of Hong Kong in 1992, where he is now Choh-Ming Li Research Professor of Information Engineering. He was the Chairman of the Department of Information Engineering from 1995 to 2003 and the Dean of the Graduate School from 2005 to 2014. He served as Science Advisor at the Innovation and Technology Commission of the HKSAR government from 2003 to 2005. He has participated in a variety of research projects on topics ranging from mobile communication, networked control to network control.

Po-Ning Chen (S'93–M'95–SM'01) received the Ph.D. degree in electrical engineering from University of Maryland, College Park, in 1994. Since 1996, he has been an Associate Professor in Department of Communications Engineering at National Chiao Tung University (NCTU), Taiwan, and was promoted to a full professor in 2001. He has served as the chairman of Department of Communications Engineering, NCTU, during 2007–2009. From 2012–2015, he was the associate chief director of Microelectronics and Information Systems Research Center, NCTU, and is now the associate dean of the College of Electrical and Computer Engineering, NCTU. Dr. Chen received the 2000 Young Scholar Paper Award from Academia Sinica, Taiwan. His research interests generally lie in information and coding theory, large deviations theory, distributed detection and sensor networks.

Yunghsiang S. Han (S'90-M'93-SM'08-F'11) was born in Taipei, Taiwan, 1962. He received B.Sc. and M.Sc. degrees in electrical engineering from the National Tsing Hua University, Hsinchu, Taiwan, in 1984 and 1986, respectively, and a Ph.D. degree from the School of Computer and Information Science, Syracuse University, Syracuse, NY, in 1993. He is now with School of Electrical Engineering & Intelligentization at Dongguan University of Technology, China. He is also a Chair Professor at National Taipei University from February 2015. His research interests are in error-control coding, wireless networks, and security. Dr. Han was a winner of the 1994 Syracuse University Doctoral Prize and a Fellow of IEEE. One of his papers won the prestigious 2013 ACM CCS Test-of-Time Award in cybersecurity.

Hanxu Hou (S'11-M'16) was born in Anhui, China, 1987. He received the B.Eng. degree in Information Security from Xidian University, Xian, China, in 2010, and Ph.D. degrees in the Dept. of Information Engineering from The Chinese University of Hong Kong in 2015 and in the School of Electronic and Computer Engineering from Peking University in 2016. He is now an Assistant Professor with the School of Electrical Engineering & Intelligentization, Dongguan University of Technology. His research interests include erasure coding and coding for distributed storage systems.