Dual Scheduling Algorithm In A Generalized Switch: Asymptotic Optimality And Throughput Optimality

Lijun Chen, Steven H. Low, and John C. Doyle

Engineering & Applied Science Division, California Institute of Technology
Pasadena, CA 91125, USA
{chen@cds., slow@, doyle@cds.}caltech.edu

Summary. Generalized switch is a model of a queueing system where parallel servers are interdependent and have time-varying service capabilities. It includes as special cases the model of data scheduling in a wireless network and the input-queued cross-bar switch model. A finite set S of queues (users) are served in discrete time by a switch. Switch state h follows a discrete-time, finite-state Markov chain. At each time slot t, the switch can choose a scheduling decision m from a finite set M, which captures the interdependency among the servers specifying which subsets of servers can be active simultaneously. Each scheduling decision has the associated vector of service rates $\tilde{r}^m(h(t))$ at which queues are served, where $h(t)$ denotes the switch state at time t.

This article considers the dual scheduling algorithm that uses rate control and queue-length based scheduling to allocate resources for a generalized switch. We first consider a saturated system in which each user has infinite amount of data to be served. We prove the asymptotic optimality of the dual scheduling algorithm for such a system, which says that the vector of average service rates of the scheduling algorithm maximizes some aggregate concave utility functions. As the fairness objectives can be achieved by appropriately choosing utility functions, the asymptotic optimality establishes the fairness properties of the dual scheduling algorithm. We next consider a system with exogenous arrivals, i.e., data flows of finite size arrive at the system randomly. For such a system, we propose a modified dual scheduling algorithm that stabilizes the system whenever the input rates are within the feasible rate region and is then throughput-optimal, i.e., achieves 100% throughput.

The dual scheduling algorithm motivates a new architecture for scheduling, in which an additional queue is introduced to interface the user data queue and the time-varying server and to modulate the scheduling process, so as to achieve different performance objectives. Further research would include scheduling with Quality of Service guarantees with the dual scheduler, and its application and implementation in various versions of the generalized switch model.

1 Introduction

We consider a general model where a set S of queues (users) are served in discrete time by a generalized switch, as defined in [24]. The generalized switch can be viewed as a discrete-time, interdependent parallel server system. The servers are interdependent in that they cannot provide service simultaneously, and the dependency among them is reflected on the constraints that specify which subsets of servers can be active at the same time. Switch state h follows a discrete-time, finite-state Markov chain. At each time slot t, the switch can choose a scheduling decision m from a finite set M, which captures the constraints imposed by the interdependency among the servers. Each scheduling decision has the associated vector of service rates $\tilde{r}^m(h(t))$ at which queues are served, where $h(t)$ is the switch state at time t.

The generalized switch model has many applications in communication networks. For example, in cellular network in the downlink, the servers correspond to the wireless links from the base stations to the users, and the constraint is that each base station can transmit to at most one of the users and each user can be served by at most one of the base stations in each time slot, see e.g. [5, 12]. Other examples include multi-hop wireless network where each wireless link can be viewed as a server and the constraints disallow simultaneous transmission of neighboring links due to interference, see e.g. [27, 28, 29]. It also includes as a single-state special case input-queued cross-bar switch where a server corresponds to each input-output port pair, and the constraint is that each input port transmits to exactly one of the output ports and each output port receives from exactly one of the input ports at any time, see e.g. [19]. The same model can extend to handle the packet switch in wireless network, where the switch state (i.e., wireless line rates) is supposedly time-varying.

For such a generalized switch system with time-varying state, the service rate that can be offered to the users (queues) is both user-dependent and time-dependent. This, on one hand, opens up the possibility to use state-aware scheduling strategies, i.e., to exploit service variations to increase the throughput. On the other hand, the parallel servers are interdependent, and to serve (schedule) always the user with the highest potential rate maximizes overall throughput but usually results in the starvation of some users. So, we need to trade off throughput for fairness. However, the time-varying nature of the generalized switch, coupled with the user-dependent service rate and unknown data arrival, makes it very challenging to design scheduling policies to fulfil fairness and throughput requirements, as well as other performance objectives.

There exists lots of work on scheduling with different performance objectives for different versions of the generalized switch model. For fair scheduling, in the context of cellular network in the downlink, one of the principal policies is the Proportional Fair Scheduler of Qualcomm High Data Rate system [5, 12], which schedules the user with the largest ratio of the current achievable rate to the exponentially smoothed throughput. This scheduling algorithm has been shown to maximize the sum of the logarithm utilities of the long-run average data rates provided to the users [30, 15, 26], and thus achieve proportional fairness [13]. The generalization of the proportional fair scheduling algorithm to any concave utility function for a generalized switch has been studied[1], see e.g. [25]. Other work on fair scheduling includes, e.g., [16, 8, 17]. For throughput-optimal scheduling that attains maximum stability region of the system, one of the principal policies is the MaxWeight scheduling in

[1] We call this type of scheduling policies the primal scheduling algorithms, since they can be seen as the gradient algorithm to solve a concave utility maximization problem directly.

the context of wireless network, see e.g. [27, 28, 29, 1, 22, 20, 10], and in the context of input-queued switch, see e.g. [19]. The stability region of a scheduling policy is the set of mean flow rate vectors such that the queue-length process is stable under this policy. The throughput-optimal scheduling has its origin in [27, 28, 29], where it is shown that allocating resources to maximize a queue-length-weighted sum of rates is a stabilizing policy under any sustainable flows. However, there is no fairness guarantee with throughput-optimal scheduling.

In this article, we study the dual scheduling algorithms for the generalized switch (see [9] for preliminary results in the context of cellular network in the downlink). These algorithms are motivated by the dual subgradient algorithm of convex optimization problems [23, 6]. With an additional queue (termed M-queue) being introduced for each user, the dual scheduling algorithm is a combination of rate control (of the M-queue) and M-queue-length based scheduling. The rate control algorithm is motivated by utility framework for TCP congestion control (see e.g. [14, 18]), which shows that various TCP congestion control protocols can be interpreted as distributed primal-dual algorithms to solve aggregate network utility maximization. The queue-length based scheduling takes the form of a simple throughput-optimal scheduling. As such, while the queue-length-based scheduling part keeps maximizing the throughput, the rate-control part modulates the scheduling process by choosing appropriate utility functions, so as to achieve various performance objectives.

Section 2 presents the details of system model and the dual scheduling algorithm for the generalized switch. In Section 3, we consider a saturated system in which each user has infinite amount of data to be served. For such a system, fairness among the users is presumably the most important concern. We present a dual scheduling algorithm, which extends the algorithm studied in [11], and prove its asymptotic optimality, which says that the vector of average service rates of the scheduling algorithm maximizes some aggregate concave utilities of the users. As is well-known, the fairness objectives can be achieved by appropriately choosing the utility functions. So, the asymptotic optimality establishes the fairness properties of the dual scheduling algorithm.

We next consider in Section 4 the system with exogenous arrivals, i.e., data flows of finite size arrive at the system randomly. For such a system, one of the performance properties of particular concern for any scheduling algorithm is its stability rate region within which the flows can be stably supported and the response time remains finite. We consider the traffic at flow level, but include packet-level dynamics, i.e., the precise operation of the scheduling mechanism. We propose another dual scheduling algorithm and show that it stabilizes the system whenever the arrival rates are within the feasible rate region and is then throughput-optimal. Compared with other throughput-optimal scheduling policies, the dual throughput-optimal scheduling algorithm provides some weighted fairness among the users at flow level.

The dual scheduling algorithm motivates a new architecture for scheduling in the generalized switch, in which an additional queue is introduced to interface the user data queue and the time-varying server and to modulate the scheduling process, so as to achieve different performance objectives such as fairness, and maximum throughput, etc. In Section 5, we will briefly discuss some implementation issues and advantages of the dual scheduler, and Quality of Service scheduling in generalized switches.

2 System Model

We consider a queueing system where a finite set S of parallel queues (users), indexed by s, are served by a generalized switch. The generalized switch can be abstracted as an interdependent parallel server system. The servers are interdependent in that they cannot provide service simultaneously, and the dependency among them is reflected on the constraints that specify which subsets of servers can be active at the same time. For convenience, we use a "dependency" graph G to capture this interdependency. Each vertex in G represents a server, and an edge between two vertices means the corresponding servers cannot be active simultaneously. Thus, only those servers in an independent set[2] of the dependency graph can be active at the same time. We denote the set of independent sets by M, with each element indexed by m.

The system operates in discrete time $t = 0, 1, 2, \cdots$. By convention, we choose the duration of a time slot as the *unit* of time, and identify time t with the unit time interval $[t, t+1)$. The switch has a finite set H of states. The switch state is fixed in one of the states $h \in H$ within a time slot but varies across slots according to an irreducible finite-state Markov chain. Corresponding to the switch state h, the service rate to user s is $r_s(h)$ packets per time slot when the switch servers only s, and the service rate vectors $\tilde{r}^m(h),\ m \in M$ that can be offered to the users are

$$\tilde{r}_s^m(h) = \begin{cases} r_s(h) & \text{if } \ s \in m \\ 0 & \text{otherwise.} \end{cases}$$

By standard time-sharing argument, the feasible rate region $\Pi(h)$ in switch state h is defined to be the convex hull of these rate vectors [4]

$$\Pi(h) := \left\{ \tilde{r} : \tilde{r} = \sum_{i=1}^{M} t_i \tilde{r}^i(h),\ t_i \geq 0,\ \sum_{i=1}^{M} t_i = 1 \right\}, \tag{1}$$

where we slightly abuse the notation and let M also denote the size of the set M. Let the switch state distribution be $d(h)$, we further define the mean feasible rate (capacity) region as

$$\overline{\Pi} = \left\{ \overline{r} : \overline{r} = \sum_{h \in H} d(h)\tilde{r}(h), \tilde{r}(h) \in \Pi(h) \right\}. \tag{2}$$

This mean rate region is a closed convex set, and is the best feasible rate region the system can support on average.

[2] An independent set of vertices is defined as a set of vertices that have no edges between each other. An empty set is an interdependent set.

2.1 Queue Length Dynamics

Fig.1 shows the architecture of the dual scheduler from the perspective of one user. The system keeps separate *data queues* for the users to buffer the data intended to them. In addition, another queue, called M-queue, is introduced for each user. The M-queue interfaces the data queue and the time-varying server, in that the data will depart from the data queue to enter the M-queue, and the server will directly serve the M-queue.

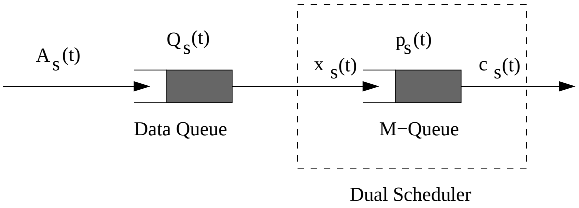

Fig. 1. The Architecture of the Dual Scheduler.

Denote the size of the data queue and M-queue for user s at the beginning of the time slot t by $Q_s(t)$ and $p_s(t)$ respectively, the number of arrivals to the data queue and M-queue of user s in time slot t by $A_s(t)$ and $x_s(t)$ respectively, and the amount of service offered to the M-queue of user s in time slot t by $c_s(t)$. The evolutions of the data queue and M-queue length for user s are given by

$$Q_s(t+1) = Q_s(t) + A_s(t) - x_s(t) \ , \tag{3}$$

$$p_s(t+1) = [p_s(t) + x_s(t) - c_s(t)]^+, \tag{4}$$

where '$+$' denotes the projection onto the set $\Re^+$ of non-negative real numbers.

We further introduce a small parameter $\gamma > 0$, and for convenience, define a new quantity $q_s(t) = \gamma p_s(t)$ for each user s. In Section 3 we will see that γ characterizes the asymptotic optimality and fairness of the dual scheduling algorithm. We call q the scaled queue-length, since it is the M-queue length scaled by γ. By equation (4), the evolution of the scaled queue-length is given by

$$q_s(t+1) = [q_s(t) + \gamma(x_s(t) - c_s(t))]^+. \tag{5}$$

With the dual scheduling algorithm, the system controls the arrival rate into the M-queues and determines service rates offered to the M-queues based on queue-length.

2.2 Dual Scheduling Algorithm

We assume that each user s attains a utility $U_s(x_s)$ when its arrival rate to the M-queue is x_s packets per time slot. $U_s(\cdot)$ may be dependent of data queue size Q_s, but is assumed to be continuously differentiable, increasing and strictly concave with respect to x_s. In time slot t, given the current M-queue length $p_s(t)$, the maximal arrival rate to the M-queue of user s is specified as following

$$x_s(t) = \min\left\{{U_s'}^{-1}(\gamma p_s(t)), \alpha_s\right\} = \min\left\{{U_s'}^{-1}(q_s(t)), \alpha_s\right\}, \tag{6}$$

where $\alpha_s > \max_h r_s(h)$ is the upper bound specified on the arrival rate, and thus $x_s(t)$ maximizes $U_s(x_s) - q_s x_s$ over $0 \le x \le \alpha_s$. Note that we choose packet of equal length as the unit of data. x_s will be rounded to closest integer automatically.

We now consider service allocation. In time slot t, given the current M-queue length $p(t)$, the switch selects a (physical) service rate vector[3]

$$c(t) \in \text{arg} \max_{c \in \Pi(h(t))} p(t)^T c = \text{arg} \max_{c \in \Pi(h(t))} q(t)^T c\,, \tag{7}$$

where we will always pick an extreme point maximizer[4]. Equation (7) takes the form of simple throughput-optimal scheduling as proposed in [27, 28], which schedules the transmissions dynamically based only on current system backlog and switch state.

Equations (3)-(7) define the dual scheduling algorithm. When the M-queue length process is stable, x_s *will be the service rate offered to user* s. This scheduling algorithm can be seen as motivated by the dual subgradient algorithm of concave maximization problem $\max_x \sum_s U_s(x_s)$, and is a combination of rate control [14, 18] and queue-length-based scheduling. As the queue-length-based scheduling part keeps maximizing the throughput, the rate-control part modulates the scheduling process by choosing appropriate utility functions, so as to achieve various performance objectives.

Given a scheduling algorithm, two of important issues that need to be addressed are to characterize its fairness property and its stability region. The fairness property governs the resource allocation among the competing users, and the stability region determines the efficiency of the scheduling algorithm as a whole. We will study them in the next two sections, respectively.

3 Asymptotic Optimality and Fairness

In this section, we consider a saturated system in which each user has infinite amount of data to be served, i.e., the user data queue is infinitely backlogged. So, the data queue is irrelevant and the choice of utility function $U_s(\cdot)$ is independent of Q_s. We will show that the dual scheduling algorithm maximizes some aggregate concave utilities and establish its fairness properties through its asymptotic optimality.

[3] We call the service rate allocated to the M-queue the physical service rate, in order to distinguish from the service rate received by the user data queue which will be x_s if M-queue is table.

[4] A point in a convex set is an extreme point if it cannot be written as a convex combination of other points in the convex set.

3.1 An Ideal Reference System

Before preceding, let us first define an ideal reference system problem,

$$\max_{x_s \geq 0, c_s \geq 0} \sum_s U_s(x_s) \tag{8}$$

$$\text{subject to } \; x \;\leq\; c \,\&\, c \;\in\; \overline{\Pi}. \tag{9}$$

The first constraint says that the arrival rate to the M-queues should not exceed the physical service rate. The second constraint says that the physical service rate should be in the mean rate region, which is the best feasible rate region the system can support. We will characterize the performance of the dual scheduling algorithm with respect to this reference system.

Proposition 1. *The solution x^* to problem (8)-(9) exists and is unique.*

Proof. The proof is trivial, since the objective function is strictly concave and the constraint set is a closed, convex set [6].

Consider the dual problem of the reference system problem (8)-(9)

$$\min_{u \geq 0} D(u) \tag{10}$$

with partial dual function

$$D(u) = \max_{x_s \geq 0, c_s \geq 0} \sum_s U_s(x_s) - u^T\,(x - c) \tag{11}$$

$$\text{subject to } \; c \;\in\; \overline{\Pi}, \tag{12}$$

where we relax only the constraint $x \leq c$ by introducing Lagrange multiplier u.

Proposition 2. *The solution u^* to dual problem (10) exists. Moreover, there is no dual gap between the primal problem (8)-(9) and the dual problem (10).*

Proof. The proof is trivial, since problem (8)-(9) is a convex optimization problem [6].

Having established the properties of the ideal reference system problem and its dual, in the next subsection we will characterize the dual scheduling algorithm with respect to them.

Remark 1: Roughly speaking, the primal scheduling algorithm is a scheduling policy whose vector of average service rates solving the problem

$$\max_{x_s \geq 0} \sum_s U_s(x_s)$$

$$\text{subject to } \; x \;\in\; \overline{\Pi}.$$

This problem is equivalent to problem (8)-(9), since mathematically c can be seen as an auxiliary variable. The primal scheduling algorithm can be seen as being motivated by the gradient algorithm to solve this problem [25], while the dual scheduling algorithm can be seen as being motivated by the dual gradient algorithm to solve the same problem.

3.2 Stochastic Stability

Note that M-queue length $p(t)$ (and scaled queue-length $q(t)$) evolves according to a discrete-time, discrete-space Markov chain. We first show that this Markov chain is stable, i.e., the queue-length process reaches a steady state and does not go unbounded to infinity. It is easy to check that the Markov chain has a countable state space, but is not necessarily irreducible. In such a general case, the state space is partitioned in transient set T and different recurrent classes R_i. We define *the system to be stable if all recurrent states are positive recurrent and the Markov process hits the recurrent states with probability one* [27]. This will guarantee that the Markov chain will be absorbed/reduced into some recurrent class, and the positive recurrence ensures the ergodicity of the Markov chain over this class.

Theorem 1. *The Markov chains described by equations (4) and (5) are stable.*

Proof. Consider the the Lyapunov function $V(q) = \|q - u^*\|_2^2$. By equations (5)-(7) and define $g(q) = c(q) - x(q)$, we have

$$\begin{aligned}
E[\Delta V_t(q)|q] &= E[V(q(t+1)) - V(q(t)) \mid q(t) = q] \\
&= E[V([q(t) - \gamma g(q(t))]^+) - V(q(t)) \mid q(t) = q] \\
&\le E[V(q(t) - \gamma g(q(t))) - V(q(t)) \mid q(t) = q] \\
&= E[-\gamma g(q(t))^T (2(q(t) - u^*) - \gamma g(q(t))) \mid q(t) = q] \\
&= 2\gamma \overline{g}(q)^T (u^* - q) + \gamma^2 E[\|g(q(t))\|_2^2 \mid q(t) = q] \\
&\le 2\gamma \overline{g}(q)^T (u^* - q) + \gamma^2 G^2,
\end{aligned}$$

where G is the upper bound of the norm of $g(q(t))$, and

$$\overline{g}(q) = \overline{c}(q) - x(q) \ \text{ with } \ \overline{c}(q) \in \arg\max_{c \in \overline{\Pi}} q^T c \,. \tag{13}$$

It is easy to check that $\overline{g}(q)$ is a subgradient[5] of the dual function $D(q)$ at point q, thus

$$\overline{g}(q)^T (u^* - q) \le D(u^*) - D(q) \,.$$

So,

$$E[\Delta V_t(q)|q] \le 2\gamma (D(u^*) - D(q)) + \gamma^2 G^2.$$

Note that $D(q)$ is a continuous function. Let

$$\delta = \max_{D(q) - D(u^*) \le \gamma G^2} \|q - u^*\|_2$$

and define $\mathcal{A} = \{q : \ \|q - u^*\|_2 \le \delta\}$. We can get

[5] Given a convex function $f \ : \mathcal{R}^n \mapsto \mathcal{R}$, a vector $d \in \mathcal{R}^n$ is a subgradient of f at a point $u \in \mathcal{R}^n$ if $f(v) \ge f(u) + (v - u)^T d, \ v \in \mathcal{R}^n$ [23, 6].

$$E[\Delta V_t(q)|q] \le -\gamma^2 G^2 \mathcal{I}_{q\in\mathcal{A}^c} + \gamma^2 G^2 \mathcal{I}_{q\in\mathcal{A}}\ ,$$

where $\mathcal{I}$ is the index function. Thus, by *Theorem 3.1* in [27], which is a trivial extension of Foster's criterion for irreducible chain [3], the Markov chain $q(t)$ is stable. Since the M-queue length $p(t) = \gamma q(t)$, the Markov chain $p(t)$ is also stable.

The above proof shows that the distance to the optimal u^* has negative conditional mean drift for all scaled queue-length that have sufficiently large distance to u^*, and implies that the scaled queue-length will stay near u^* when γ is small enough.

Remark 2: We can make the Markov chain $p(t)$ irreducible over its state space, by making the arrival $x(t)$ a random variable with mean $\min\{{U'_s}^{-1}(\gamma p_s(t)), \alpha_s\}$, as that assumed in [11]. We can also make the system to reach a specific state infinitely often with finite mean recurrence times, which will ensure that the system reduces to one recurrent class whatever the initial state is.

3.3 Asymptotic Optimality and Fairness

In this subsection, we will prove the asymptotic optimality of the dual scheduling algorithm in terms of dual and primal functions of the reference system problem (8)-(9).

Theorem 2. *The dual scheduling algorithm (4)-(7) converges statistically to within a small neighborhood of the optimal value $D(u^*)$, i.e.,*

$$D(u^*) \le D(E[q(\infty)]) \le D(u^*) + \gamma G^2/2, \tag{14}$$

where $q(\infty)$ is a notation used to denote the state of the Markov chain $q(t)$ in the steady state.

Proof. The first inequality $D(u^*) \le D(p)$ always holds, since $D(u^*)$ is the minimum of the dual function $D(u)$.

Now we prove the second inequality. From the proof of Theorem 1, we have

$$\begin{aligned} E[\Delta V_t(q)|q] &= E[V(q(t+1)) - V(q(t)) \mid q(t) = q] \\ &\le 2\gamma(D(u^*) - D(q)) + \gamma^2 G^2. \end{aligned}$$

Taking expectation over q, we get

$$\begin{aligned} E[\Delta V_t(q)] &= E[V(q(t+1)) - V(q(t))] \\ &\le 2\gamma(D(u^*) - E[D(q)]) + \gamma^2 G^2. \end{aligned}$$

Taking summation from $\tau = 0$ to $\tau = t-1$, we obtain

$$E[V(q(t))] \le E[V(q(0))] - 2\gamma \sum_{\tau=0}^{t-1} E[D(q(\tau))] + 2\gamma t D(u^*) + t\gamma^2 G^2.$$

Since $E[V(q(t))] \geq 0$, we have

$$2\gamma \sum_{\tau=0}^{t-1} E[D(q(\tau))] - 2\gamma t D(u^*) \leq E[V(q(0))] + t\gamma^2 G^2.$$

From this inequality we obtain

$$\frac{1}{t} \sum_{\tau=0}^{t-1} E[D(q(\tau))] - D(u^*) \leq \frac{E[V(q(0))] + t\gamma^2 G^2}{2t\gamma}.$$

Note that $q(t)$ is stationary and ergodic in some steady state by Theorem 1, and so is $D(q(t))$. Thus,

$$\lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} E[D(q(\tau))] = E[D(q(\infty))].$$

So,

$$E[D(q(\infty))] - D(u^*) \leq \gamma G^2/2.$$

Since $D(q)$ is a convex function, by Jensen's inequality,

$$D(E[q(\infty)]) - D(u^*) \leq \gamma G^2/2,$$

i.e., the algorithm converges statistically to within $\gamma G^2/2$ of the optimal value $D(u^*)$.

Since $D(q)$ is a continuous function, Theorem 2 implies that the scaled queue-length q approaches u^* statistically when γ is small enough.

Corollary 1. *$x(t)$ is a stable Markov chain. Moreover, the average arrival rates $E[x(\infty)] \in \overline{\Pi}$, where $x(\infty)$ denotes the state of the process $x(t)$ in the steady state.*

Proof. $x(t)$ is a deterministic, finite-value function of $q(t)$. $x(t)$ is a stable Markov chain, since $q(t)$ is. $E[x(\infty)] \in \overline{\Pi}$, otherwise the average scaled queue-length $E[q(\infty)]$ will go unbounded, which contradicts to Theorem 2.

Theorem 3. *Let $P(x)$ be the primal function of the reference system problem (8)-(9). The dual scheduling algorithm (4)-(7) converges statistically to within a small neighborhood of the optimal value $P(x^*)$, i.e.,*

$$P(x^*) \geq P(E[x(\infty)]) \geq P(x^*) - \frac{\gamma G^2}{2}\ . \tag{15}$$

Proof. The first inequality $P(x^*) \geq P(E[x(\infty)])$ holds, since $E[x(\infty)] \in \overline{\Pi}$.

Now we prove the second inequality. By equation (5), we have

$$
\begin{aligned}
& E[||q(t+1)||_2^2|q(t)] \\
= & E[||[q(t)-\gamma g(q(t))]^+||_2^2|q(t)] \\
\leq & E[||q(t)-\gamma g(q(t))||_2^2|q(t)] \\
= & ||q(t)||_2^2 - 2\gamma \overline{g}(q(t))^T q(t) + \gamma^2 E[||g(q(t))||_2^2|q(t)] \\
= & ||q(t)||_2^2 + 2\gamma \sum_s U_s(x_s(t)) - 2\gamma \sum_s (U_s(x_s(t)) - q_s(t)x_s(t)) \\
& -2\gamma \sum_s q_s(t)\overline{c}_s(t) + \gamma^2 E[||g(p(t))||_2^2|q(t)] \\
\leq & ||q(t)||_2^2 + 2\gamma \sum_s U_s(x_s(t)) - 2\gamma \sum_s (U_s(x_s^*) - q_s(t)x_s^*) \\
& -2\gamma \sum_s q_s(t)\overline{c}_s(t) + \gamma^2 E[||g(p(t))||_2^2|q(t)] \\
= & ||q(t)||_2^2 + 2\gamma P(x(t)) - 2\gamma P(x_s^*) \\
& -2\gamma \sum_s q_s(t)(\overline{c}_s(t) - x_s^*) + \gamma^2 E[||g(q(t))||_2^2|q(t)] \\
\leq & ||q(t)||_2^2 + 2\gamma P(x(t)) - 2\gamma P(x^*) + \gamma^2 E[||g(q(t))||_2^2|q(t)] \\
\leq & ||q(t)||_2^2 + 2\gamma P(x(t)) - 2\gamma P(x^*) + \gamma^2 G^2,
\end{aligned}
$$

where $\overline{g}(q)$ is defined as in equation (13), the second inequality follows from the fact that $x_s(t)$ is the maximizer of $\max_{x_s}(U_s(x_s) - q_s x_s)$, and the third inequality follows from the fact that $\overline{c}(t)$ is the maximizer in equation (13) and $x^* \in \overline{\Pi}$.

Taking expectation over q, we get

$$E[||q(t+1)||_2^2] \leq E[||q(t)||_2^2] + 2\gamma E[P(x(t))] - 2\gamma P(x^*) + \gamma^2 G^2.$$

Applying the inequalities recursively, we obtain

$$E[||p(t)||_2^2] \leq E[||p(0)||_2^2] + 2\gamma \sum_{\tau=0}^{t-1} (E[P(x(\tau))] - P(x^*)) + t\gamma^2 G^2.$$

Since $E[||p(t)||_2^2] \geq 0$, we have

$$2\gamma \sum_{\tau=0}^{t-1} (E[P(x(\tau))] - P(x^*)) \geq -E[||p(0)||_2^2] - t\gamma^2 G^2.$$

From this inequality we obtain

$$\frac{1}{t} \sum_{\tau=0}^{t-1} E[P(x(\tau))] - P(x^*) \geq \frac{-E[||p(0)||_2^2] - t\gamma^2 G^2}{2t\gamma}.$$

Note that $x(t)$ is stationary and ergodic in some steady state by Corollary 1. Thus,

$$\lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} E[P(x(\tau))] = E[P(x(\infty))].$$

So,

$$E[(P(x(\infty))] - P(x^*) \geq -\frac{\gamma G^2}{2}.$$

Since P is a concave function, by Jenson's inequality,

$$P(E[x(\infty)]) - P(x^*) \geq -\frac{\gamma G^2}{2},$$

i.e., the algorithm converges statistically to within $\gamma G^2/2$ of the optimal value $P(x^*)$.

Since $P(x)$ is a continuous function, Theorems 3 implies that the average arrival rates to the M-queues approaches the optimal of the ideal reference system (8)-(9) when γ is small enough. Note that, when the M-queue length is stable, x will be the service rates offered to the users. Theorem 2 and 3 shows that, surprisingly, the vector of average service rates offered by the dual scheduling algorithm (4)-(7) approximately solves the ideal reference system problem, which is to maximize the aggregate concave utilities over the best feasible rate region that the network can support.

As is well-known, the fairness objectives can be achieved by appropriately choosing the concave objective functions. So, the asymptotic optimality establishes the fairness properties of the dual scheduling algorithm. For example, if we choose logarithm utility function $U_s(x_s) = \log(x_s)$, the dual scheduler will achieve proportional fairness [13, 11].

4 Throughput-optimal Scheduling

The results presented in last section assume a saturated system in which each user has infinite amount of data to be served. In reality, however, data flows of finite size arrive at the system for each user according to some arrival process. For such a system, one of the properties of particular concern for a scheduling algorithm is its stability rate region within which the flows can be stably supported and the response time remains finite, since it characterizes the throughput performance of the scheduling algorithm as a whole. In this section, we will consider a scheduling policy which maximizes the system throughput while providing some weighted fairness among users at flow level.

We will model the traffic in the system at the level of flows. We assume that the arrival process of data flows for each user s is a stationary point process with intensity ν_s, and the size of the flows to be served is drawn independently from a general distribution with mean $1/\mu_s$ and finite second moment. We denote by $\rho_s = \nu_s/\mu_s$ the traffic intensity offered to user s. The data queue length (unfinished work) $Q(t)$ is a stochastic process which increases as new flows come and decreases as date is transferred into the M-queues. We are concerned with the stability of $Q(t)$, and derive conditions on the traffic intensity ρ for which, starting from any initial state, the amount of data (or number of the flows) to be served remains finite with probability 1. Obviously, the following condition is necessary

$$\rho \in \text{int}(\overline{\Pi}), \tag{16}$$

where int$(\overline{\Pi})$ denotes the interior of convex set $\overline{\Pi}$. This condition for stability is not sufficient for general scheduling policies. We call a scheduling algorithm *throughput-optimal* if condition (16) is also sufficient under this algorithm.

4.1 Throughput Optimality and Fairness

We would like the dual scheduling algorithm discussed in last section to be a throughput-optimal policy. However, fairness and maximum throughput are not directly compatible performance objectives. On one hand, as we can see in e.g. [27, 28, 29], there is no guarantee of fairness with throughput-optimal scheduling. On the other hand, fair scheduling may not be stable for some flows within the feasible rate region, see e.g. the discussion in [2] for proportional fair scheduling algorithm in cellular network in the downlink. The main reason why the fair scheduling may not be a stabilizing policy is because it is usually traffic-independent. For example, in proportional fair scheduling algorithm, the scheduling decision is made based on the current achievable rate and the mean throughput but not on the real traffic intended to the users.

Given the above consideration, a throughput-optimal scheduling policy should be traffic-dependent. This motivates us to consider scheduling algorithm whose vector of service rates x solves the following utility maximization

$$\max_{x_s \geq 0, c_s \geq 0} \sum_s Q_s U_s(x_s) \tag{17}$$

$$\text{subject to } \; x \leq c \,\&\, c \in \overline{\Pi}, \tag{18}$$

where $U_s(\cdot)$ is again independent of the data queue length Q_s. However, we cannot solve problem (17)-(18) directly, since the system only knows the current switch state but not its statistics.

By the assumptions about the data flows for each user, the arrival to the data queues $A_s(t)$ is i.i.d. across time slots and with mean $\rho_s = E[A_s(t)]$ and finite second moment. The evolution of unfinished work $Q(t)$ and the modified dual scheduling algorithm are given by,

$$Q_s(t+1) = [Q_s(t) + A_s(t) - x_s(t)]^+, \tag{19}$$

$$q_s(t+1) = [q_s(t) + \gamma(\tilde{x}_s(t) - c_s(t))]^+, \tag{20}$$

$$x_s(t) = \min\{{U'_s}^{-1}(q_s(t)/Q_s(t)), \alpha_s\}, \tag{21}$$

$$c(t) \in \arg\max_{c \in \Pi(h(t))} q(t)^T c\,, \tag{22}$$

where $\tilde{x}_s(t) = \min\{Q_s(t) + A_s(t), x_s(t)\}$. Equation (20)-(22) is the modified dual scheduling algorithm motivated by the utility maximization (17)-(18). It makes scheduling decision based only on current backlog and switch state. Note that, if the unfinished work $Q(t)$ freezes, this algorithm will solve the utility maximization approximately.

4.2 Optimality Proof

It is easy to verify that the pair $(Q(t), q(t))$[6] evolves according to a discrete-time, discrete-space Markov chain. The following theorem proves the stability of this Markov chain under any sustainable flows, and show that the modified dual scheduling algorithm is a throughput-optimal scheduling policy.

Theorem 4. *The Markov chain $Y(t) = (Q(t), q(t))$ is stable iff traffic condition (16) is satisfied.*

Before we prove Theorem 4, let first introduce two lemmas. Consider Lyapunov function

$$V(Y) = V_1(Y) + V_2(Y),$$

where $V_1(Y) = \sum_s b_s Q_s^2 = \sum_s U_s'((1+\epsilon)\rho_s)Q_s^2$, and $V_2(Y) = \sum_s \frac{q_s^2}{\gamma}$. ϵ is a small positive constant that will be decided later on. Note that $b_s = U_s'((1+\epsilon)\rho_s) > 0$, since $U_s(\cdot)$ is an increasing function. Also, note that $U_s'(\cdot)$ is a decreasing function.

Lemma 1. *For any $\rho \in int(\overline{\Pi})$, there exists a $\epsilon > 0$ such that the following inequality holds,*

$$\begin{aligned} &E[V_1(Y(t+1)) - V_1(Y(t))|Y(t)] \\ &\leq -2\epsilon \sum_s b_s \rho_s Q_s(t) + 2\sum_s q_s(t)((1+\epsilon)\rho_s - x_s(t)) + T_1, \end{aligned}$$

where T_1 is a positive constant.

Proof. For any $\rho \in \text{int}(\overline{\Pi})$, there exists a small number $\epsilon > 0$ such that $(1+\epsilon)\rho \in \text{int}(\overline{\Pi})$, since $\overline{\Pi}$ is a closed convex set and $0 \in \overline{\Pi}$. By equation (19), we have

$$\begin{aligned} &E[V_1(Y(t+1)) - V_1(Y(t))|Y(t)] \\ &= \sum_s b_s E[([Q_s(t) + A_s(t) - x_s(t)]^+)^2 - Q_s^2(t)|Y(t)] \\ &\leq \sum_s b_s E[(Q_s(t) + A_s(t) - x_s(t))^2 - Q_s^2(t)|Y(t)] \\ &= \sum_s b_s E[2Q_s(t)(A_s(t) - x_s(t))|Y(t)] + \sum_s b_s E[(A_s(t) - x_s(t))^2|Y(t)] \end{aligned}$$

[6] Throughout this section, for convenience, we will use scaled quene-length $q(t)$ rather than the M-queue length $p(t)$ to establish the stability and optimality properties of the dual scheduling algorithm. This makes no difference, since $q(t)$ is $p(t)$ scaled by a constant and all the properties of $q(t)$ carry to $p(t)$ and vice versa.

$$\begin{aligned}
&\leq \sum_s 2b_s Q_s(t)(\rho_s - x_s(t)) + T_1 \\
&= -\sum_s 2\epsilon b_s \rho_s Q_s(t) + \sum_s 2b_s Q_s(t)((1+\epsilon)\rho_s - x_s(t)) + T_1 \ . \qquad (23)
\end{aligned}$$

Here the positive constant T_1 is the upper bound of $\sum_s b_s E[(A_s(t) - x_s(t))^2 | Y(t)]$. This can be achieved since $x_s(t)$ is upper bounded by α_s and $E[A_s^2(t)]$ is also bounded.

Now we consider the second term in (23). If $(1+\epsilon)\rho_s \geq x_s(t)$,

$$\begin{aligned}
&2b_s Q_s(t)((1+\epsilon)\rho_s - x_s(t)) \\
&\leq 2Q_s(t)U_s'(x_s(t))((1+\epsilon)\rho_s - x_s(t)) \\
&= 2q_s(t)((1+\epsilon)\rho_s - x_s(t)),
\end{aligned}$$

where the first inequality follows from the fact that $U_s'(\cdot)$ is a decreasing function, and the last equality follows from the fact $x_s(t) \leq (1+\epsilon)\rho_s < \alpha_s$ and equation (21). If $(1+\epsilon)\rho_s < x_s(t)$,

$$\begin{aligned}
&2b_s Q_s(t)((1+\epsilon)\rho_s - x_s(t)) \\
&= 2q_s(t)((1+\epsilon)\rho_s - x_s(t)) + 2(b_s Q_s(t) - q_s(t))((1+\epsilon)\rho_s - x_s(t)) \\
&\leq 2q_s(t)((1+\epsilon)\rho_s - x_s(t)) + 2(Q_s(t)b_s - Q_s(t)U_s'(x(t)))((1+\epsilon)\rho_s - x_s(t)) \\
&\leq 2q_s(t)((1+\epsilon)\rho_s - x_s(t)),
\end{aligned}$$

where the first inequality follows from the equation (21), and the last inequality follows from the fact that $U_s'(\cdot)$ is a decreasing function.

So, in any case, we get

$$\begin{aligned}
&E[V_1(Y(t+1)) - V_1(Y(t)) | Y(t)] \\
&\leq -2\epsilon \sum_s b_s \rho_s Q_s(t) + 2\sum_s q_s(t)((1+\epsilon)\rho_s - x_s(t)) + T_1 \ .
\end{aligned}$$

Lemma 2.

$$E[V_2(Y(t+1)) - V_2(Y(t)) | Y(t)] \leq 2\sum_s q_s(t)(x_s(t) - \overline{c}_s(t)) + T_2 \ ,$$

where T_2 is a positive constant, and $\overline{c}(t) =$ arg $\max_{c \in \overline{\Pi}} q(t)^T c$.

Proof. By equation (20), we have

$$\begin{aligned}
&E[V_2(Y(t+1)) - V_2(Y(t)) | Y(t)] \\
&= \sum_s \frac{1}{\gamma} E[([q_s(t) + \gamma(\tilde{x}_s(t) - c_s(t))]^+)^2 - q_s^2(t) | Y(t)] \\
&\leq \sum_s \frac{1}{\gamma} E[(q_s(t) + \gamma(x_s(t) - c_s(t)))^2 - q_s^2(t) | Y(t)]
\end{aligned}$$

$$
\begin{aligned}
&= 2\sum_s E[q_s(t)(x_s(t)-c_s(t))|Y(t)] + \sum_s \gamma E[(x_s(t)-c_s(t))^2|Y(t)] \\
&= 2\sum_s q_s(t)(x_s(t)-\overline{c}_s(t)) + \sum_s \gamma E[(x_s(t)-c_s(t))^2|Y(t)] \\
&= 2\sum_s q_s(t)(x_s(t)-\overline{c}_s(t)) + T_2\ .
\end{aligned}
$$

Here the positive constant T_2 is the upper bound of $\sum_s \gamma E[(x_s(t)-c_s(t))^2|Y(t)]$. This can be achieved since both $x_s(t)$ and $c_s(t)$ are bounded.

With Lemmas 1 and 2, we are ready to prove Theorem 4.

Proof. For any $\rho \in \text{int}(\overline{\Pi})$, there exists a small number ϵ such that $(1+\epsilon)\rho, (1+2\epsilon)\rho \in \text{int}(\overline{\Pi})$. Thus, by Lemmas 1 and 2, for any arrival process $A(t)$ with mean ρ we have

$$
\begin{aligned}
&E[V(Y(t+1)) - V(Y(t))|Y(t)] \\
\le& -2\epsilon\sum_s b_s\rho_s Q_s(t) + 2\sum_s q_s(t)((1+\epsilon)\rho_s - x_s(t)) + T_1 \\
&+2\sum_s q_s(t)(x_s(t)-\overline{c}_s(t)) + T_2 \\
=& -2\epsilon\sum_s b_s\rho_s Q_s(t) + 2\sum_s q_s(t)((1+\epsilon)\rho_s - \overline{c}_s(t)) + T \\
=& -2\epsilon\sum_s b_s\rho_s Q_s(t) - 2\epsilon\sum_s \rho_s q_s(t) + T \\
&+2\sum_s q_s(t)((1+2\epsilon)\rho_s - \overline{c}_s(t)) \\
\le& -2\epsilon\sum_s b_s\rho_s Q_s(t) - 2\epsilon\sum_s \rho_s q_s(t) + T,
\end{aligned}
$$

where $T = T_1 + T_2$, and the last inequality follows from the fact that $\overline{c}(t)$ maximizes $q(t)^T c$ over $c \in \overline{\Pi}$.

We can see that when $||Y(t)||_2$ is large enough, the conditional mean drift of Lyapunov function $V(Y(t))$ is negative. Thus, by *Theorem 3.1* in [27], the Markov chain $Y(t)$ is stable.

Theorem 4 shows that the system is stable under any sustainable flow $\rho \in \text{int}(\overline{\Pi})$. So, the modified dual scheduling algorithm is a throughput-optimal policy. Note that we prove the throughput optimality under a very general assumption of the flow arrival process, while most similar results assume a Poisson arrival with exponentially distributed file size. Also, the above stability result is independent of the value of parameter γ, while in section 3 the parameter γ characterizes the optimality of the dual scheduling algorithm.

4.3 Flows with Exponentially Distributed Size

If the flow lengths are exponentially distributed with parameter $1/\mu_s$, the system can be modelled by a Markov chain $(n(t), q(t))$, where $n_s(t)$ is the number of active data flows for user s by the beginning of time slot t. Similarly, we would like the vector of service rates x provided by the scheduling algorithm solves the following utility maximization

$$\max_{x_s \geq 0, c_s \geq 0} \sum_s n_s U_s(x_s) \tag{24}$$

$$\text{subject to} \quad x \leq c \,\&\, c \in \overline{\Pi}. \tag{25}$$

Let $a_s(t)$ denote the number of new flows of user s arriving in time slot t, and $d_s(t)$ the number of flows that complete data transfer in time slot t. Thus, $E[a_s(t)] = \nu_s$ and $E[d_s(t)|n(t), q(t)] = \mu_s x_s(t)$. Motivated by the dual subgradient algorithm for the above utility maximization, the evolution of $n(t)$ and the modified dual scheduling algorithm are given by

$$n_s(t+1) = n_s(t) + a_s(t) - d_s(t)\ , \tag{26}$$

$$q_s(t+1) = [q_s(t) + \gamma(x_s(t) - c_s(t))]^+, \tag{27}$$

$$x_s(t) = \min\{{U'_s}^{-1}(q_s(t)/n_s(t)), \alpha_s\}\ , \tag{28}$$

$$c(t) \in \arg \max_{c \in \Pi(h(t))} q(t)^T c\ . \tag{29}$$

Considering Lyapunov function $V(n, q) = \sum_s b_s n_s^2/\mu_s + \sum_s q_s^2/\gamma$ and following the same procedure as that in the proof of Theorem 4, we can directly obtain the following theorem.

Theorem 5. *The Markov chain $(n(t), q(t))$ is stable iff traffic condition (16) is satisfied.*

This theorem shows that the system is stable under any sustainable flow $\rho \in$ int$(\overline{\Pi})$. So, the modified dual scheduling algorithm (27)-(29) is a throughput-optimal policy. Compared to other throughput-maximum policy, the modified dual scheduling algorithm has the advantage of providing some short-term fairness. To see this, note that equations (27)-(29) are dual subgradient algorithm. If the number of active flows n_s for each user freezes, the above scheduling algorithm will solve the utility maximization (24)-(25). But n_s is time-varying, and the system cannot achieves the asymptotic optimality before the change of n_s. Nonetheless, at the level of flows whose time-scale is large comparing to the duration of the time slot, the algorithm will provide some level of fairness, since the system evolves along the gradient direction to the optimal.

Note that throughout this section, we consider the system traffic at flow level, but include packet level dynamics, i.e., the precise operation of scheduling mechanism.

We may also assume “the separation of time-scale”[7], i.e., the modified dual scheduling algorithm (27)-(29) works at fast time scale and achieves resource sharing objective (24)-(25) perfectly at flow level. Under this assumption, $n(t)$ is an irreducible Markov chain, and we can choose Lyapunov function $V(n(t)) = \sum_s b_s n_s^2/\mu_s$ to prove that the algorithm is a throughput-optimal policy.

Theorem 6. *Under the assumption of the separation of time-scale, the Markov chain $n(t)$ is stable iff traffic condition (16) is satisfied.*

Proof. Let x denote the maximizer of the problem (24)-(25). For any $\rho \in \text{int}(\overline{\Pi})$, there exists a small number ϵ such that $(1+\epsilon)\rho \in \text{int}(\overline{\Pi})$. So, $\sum_s n_s U_s(x_s) \geq \sum_s n_s U_s((1+\epsilon)\rho_s)$. Note that $U_s(\cdot)$ is a strictly concave function, thus

$$\begin{aligned} &\sum_s n_s b_s (x_s - (1+\epsilon)\rho_s) \\ &\geq \sum_s n_s U_s(x_s) - \sum_s n_s U_s((1+\epsilon)\rho_s) \geq 0 \, . \end{aligned} \tag{30}$$

Now consider the conditional drift of Lyapunov Function $V(n)$,

$$\begin{aligned} &E[V(n(t+1)) - V(n(t)) | n(t)] \\ &= 2\sum_s \frac{b_s}{\mu_s} E[n_s(t)(a_s(t) - d_s(t)) | n(t)] + \sum_s \frac{b_s}{\mu_s} E[(a_s(t) - d_s(t))^2 | n(t)] \\ &\leq 2\sum_s b_s n_s(t)(\rho_s - x_s) + T_3, \end{aligned}$$

where positive constant T_3 is the upper bound of $\sum_s \frac{b_s}{\mu_s} E[(a_s(t) - d_s(t))^2 | n(t)]$. Thus,

$$\begin{aligned} &E[V(n(t+1)) - V(n(t)) | n(t)] \\ &\leq -2\epsilon \sum_s b_s \rho_s n_s(t) + T_3 + 2\sum_s b_s n_s(t)((1+\epsilon)\rho_s - x_s) \\ &\leq -2\epsilon \sum_s \rho_s b_s n_s(t) + T_3 \ , \end{aligned}$$

where the last inequality follows from equation (30). Thus, by Foster’s criterion [3], the Markov chain $n(t)$ is stable.

Remark 3: Many throughput-optimal scheduling algorithms for different versions of the generalized switch model have been proposed, see e.g. [27, 28, 29, 19, 1, 22, 24, 10]. Most of these scheduling policies maximize $\sum_s f_s(Q_s(t))x_s$, where $f_s(\cdot)$ is a explicitly or implicitly-defined function of backlog Q_s, in order to achieve different performance objectives or implement different scheduling criteria. However,

[7] Due to its analytical tractability, most researches on user-level performance of scheduling algorithms assume a separation of time-scale, see e.g. [8, 7].

to our knowledge there does not exist any throughput-optimal policy which maximizes $\sum_s Q_s(t)F_s(x_s)$, where $F_s(\cdot)$ is a function of service rate x_s. It is possible to achieve different performance objectives by different choices of $F_s(\cdot)$. In the modified dual scheduling algorithm, we can say that we introduce M-queue to modulate the scheduling, which implicitly defines a function $F_s(\cdot)$. This modulation will definitely change the dynamics of the scheduling process, and to provide some weighted fairness among the users at flow level is a such consequence. It is interesting to further study the related issues.

5 A New Scheduling Architecture

The dual scheduling algorithm motivates a new architecture for scheduling in the generalized switch (please see Fig.1 for a pictorial depiction). In this new architecture, a queue, termed M-queue, is introduced to interface the user data queue and the time-varying server. Data will depart from the data queue to enter the M-queue, and the generalized switch serves directly the M-queue. Through controlling the arrival process to the M-queue (or the departure process from the data queue), we can modulate the scheduling process, in order to achieve different performance objectives such as fairness, and maximum throughput, etc.

The dual scheduler would not incur much additional complexity. M-queues are distributed at each user, and can be “virtual” or be implemented as physical queues. The control of the M-queue arrival process is also distributed at each user and depends on only the “local” queue length of each user. The dual scheduler provides some advantages over other scheduling algorithms. For example, in the cellular network in the downlink, even though the primary scheduling algorithm can achieve fair resource allocation, it requires to estimate the average throughput of the users, while with the dual algorithm we only need to simply measure the M-queue length. Also, the dynamics of the M-queue ($p(t)$ and $x(t)$) is feedback-controlled, and thus will be relatively smooth, comparing with the dynamics of the switch. So, the dual scheduler can provide a relatively reliable and smooth service to the users, and can behave as a good interface between higher layer protocols and the scheduling at the link layer and ensure a better performance of the higher-layer protocols such as that of TCP congestion control.

To provide Quality of Service in generalized switches is a difficult problem. In the context of wireless networks, the interdependence of wireless links in combination with the time-varying nature of wireless channel makes QoS scheduling fairly challenging and the available results are mostly on stability guarantees. In the context of input-queued cross-bar switch, input buffering makes scalable switch design possible but makes QoS guarantees very challenging and again most available results are on maximizing the throughput, see e.g. [21] for a review. The dual scheduler might be promising in providing QoS in generalized switches, through carefully designing the M-queue arrival process. Further study is needed on related issues.

6 Conclusions

In this article, we consider the dual scheduling algorithm for a generalized switch. For a saturated system, we prove the asymptotic optimality of the dual scheduling algorithm and thus establish its fairness properties. For a system with exogenous arrivals, we propose a modified dual scheduling algorithm, which is throughput-optimal while providing some weighted fairness among the users at the level of flows.

The dual scheduling algorithm motivates a new architecture for scheduling, in which an additional queue is introduced to interface the user data queue and the time-varying server and to modulate the scheduling process, so as to achieve different performance objectives. Further research stemming out of this article includes scheduling with Quality of Service guarantees with the dual scheduler, and its application and implementation in various versions of the generalized switch model.

Acknowledgments

This work is performed as part of the Caltech FAST Project supported by NSF, Caltech Lee Center for Advanced Networking, ARO, AFOSR, DARPA, and Cisco.

References

1. M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar and P. Whiting, Providing quality of services over shared wireless link, *IEEE Communication Magazine*, Feburary 2001.
2. M. Andrews, Instability of the proportional fair scheduling algorithm for HDR, *IEEE Transactions on Wireless Communications*, **3**(5):104-116, 2004.
3. S. Asmussen, *Applied Probability and Queues*, 2nd edition, Springer, 2003.
4. A. Bar-Noy, A. Mayer, B. Schieber and M. Sudan, Guaranteeing fair service to persistent dependent tasks, *SIAM J. COMPUT.*, **27**(4):1168-1189, August 1998.
5. P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana and A. Viterbi, CDMA/HDR: a bandwidth-efficient high-speed wireless data service for nomadic users, *IEEE Communication Magazine*, **38**(7):70-77, 2000.
6. D. Bertsekas, *Nonlinear Programming*, 2nd edition, Athena Scientific, 1999.
7. T. Bonald and A. Proutiere, Wireless downlink data channels: user performance and cell dimensioning, *Proc. ACM Mobicom*, 2003.
8. S. Borst, User-level performance of channel-aware scheduling algorithm in wireless data networks, *Proc. IEEE Infocom*, 2003.
9. L. Chen, S. H. Low and J. C. Doyle, On the dual scheduling algorithm: asymptotic optimality and throughput optimality, Submitted, 2006.
10. A. Eryilmaz, R. Srikant and J. Perkins, Stable scheduling policies for fading wireless channels, *IEEE/ACM Transactions on Networking*, 2005.
11. A. Eryilmaz and R. Srikant, Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control, *Proc. IEEE Infocom*, 2005. Submitted for journal publication.

12. A. Jalali, R. Padovani and R. Pankaj, Data-throughput of CDMA-HDR a high efficiency data rate personal communication wireless system, *Proc. 50th IEEE VTC*, 2000.
13. F. P. Kelly, Charging and rate control for elastic traffic, *European Transactions on Telecommunications*, **8**:33-37, 1997.
14. F. P. Kelly, A. K. Maulloo and D. K. H. Tan, Rate control for communication networks: Shadow prices, proportional fairness and stability, *Journal of Operations Research Society*, **49**(3):237-252, March 1998.
15. H. J. Kushner and P. A. Whiting, Convergence of proportional-fair sharing algorithms under general conditions, *IEEE/ACM Transactions on Wireless Communications*, **3**(4):1250-1259, 2004.
16. X. Liu, E. Chong and N. Shroff, Opportunistic transmission scheduling with resource-sharing constraints in wireless networks, *IEEE J. Sel. Area Comm.*, **19**(10):2053-2064, 2001.
17. Y. Liu and E. Knightly, Opportunistic fair scheduling over multiple wireless channels, *Proc. IEEE Infocom*, April 2003.
18. S. H. Low and D. E. Lapsley, Optimal flow control I: Basic algorithm and convergence, *IEEE/ACM Transactions on Networking*, **7**(6):861-874, December 1999.
19. N. McKeown, V. Anantharam and J. Walrand, Achieving 100% throughput in an inputed-queued switch, *Proc. IEEE Infocom*, 1996.
20. M. Neely, E. Modiano and C. Rohrs, Dynamic power allocation and routing for time varying wireless networks, *Proc. IEEE Infocom*, April 2003.
21. G. Nong and M. Hamdi, On the provision of Quanlity of Service guarantees for input queued switches, *IEEE Communication Magazine*, December 2000.
22. S. Shakkottai and A. Stolyar, Scheduling for multiple flows sharing a time-varying channel: the exponential rules, *Transactions of the AMS*, Series 2, A volume in memory of F. Karpelevich, 2002.
23. N. Z. Shor, *Minimization Methods for Non-Differentiable Functions*, Springer-Verlag, 1985.
24. A. L. Stolyar, MaxWeight scheduling in a generalized switch: state space collapse and workload minimization in heavy traffic, *Ann. Appl. probab.*, **14**(1):1-53, 2004.
25. A. L. Stolyar, On the asymptotic optimality of the gradient scheduling algorithm for multiuser throughput allocation, *Operation Research*, **53**(1):12-25, 2005.
26. V. Subramanian and R. Agrawal, A stochastic approximation analysis of channel condition aware wireless scheduling algorithms, *Proc. INFORMS Telecommun. Conf.*, 2002.
27. L. Tassiulas and A. Ephremides, Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks, *IEEE Transactions on Automatic Control*, **37**(12):1936-1948, December 1992.
28. L. Tassiulas and A. Ephremides, Dynamic server allocation to parallel queues with randomly varying connectivity, *IEEE Transactions on Information Theory*, **39**:466-478, 1993.
29. L. Tassiulas, Scheduling and performance limits of networks with constantly changing topology, *IEEE Transactions on Information Theory*, **43**(3):1067-1073, 1997.
30. D. N. Tse, Multiuser diversity and proportionally fair scheduling, *In preparation*, 2004.