Learning-Aided Scheduling for Mobile Virtual Network Operators with QoS Constraints

Tianxiao Zhang*, Huasen Wu*, Xin Liu*, and Longbo Huang†
*Computer Science, University of California, Davis
†IIIS, Tsinghua University

***Abstract*—Mobile Virtual Network Operators (MVNOs) serve their customers by leasing resource from physical Mobile Network Operators (MNOs). Guaranteeing service quality by connecting customers to appropriate MNOs based on their performance is important for MVNOs, but obtaining accurate statistics of performance for all MNOs is costly. In this paper, we study the scheduling problem with QoS constraints for MVNOs without *a priori* knowledge on the system statistics such as traffic and service quality. We propose a Learning-Aided Scheduling (LSchd) algorithm based on Lyapunov optimization approaches. We show that LSchd achieves near-optimal network utility subject to average QoS constraints. Further, we propose a Dual-Learning-Aided Scheduling (DSchd) algorithm to accelerate the convergence speed. The proposed algorithms are evaluated by simulations based on real network traces. The simulation results show that even when the system statistics are non-stationary, the proposed algorithms achieve near-optimal utilities and the DSchd algorithm quickly approaches the near-optimal performance.**

I. Introduction

Mobile Virtual Network Operator (MVNO) is becoming increasingly popular over the last decade, and is attracting more attention from Internet service providers and equipment manufacturers such as Google and Huawei [1–3]. For example, Google announced Project Fi in April 2015, which is essentially a virtual operator using T-Mobile, Sprint, and WiFi hotspots all over the country to offer users the most appropriate connection at a given moment [2]. A recent report [4] predicts that global MVNO subscribers are expected to exceed 300 million by 2020, at a compound average growth rate (CAGR) of 10.7% from 2014 to 2020, and the global MVNO market is expected to reach USD 73 billion by 2020.

In addition to lower price and wider coverage-range, one important advantage of MVNOs is that they can provide services with specific quality guarantees by leasing wireless network resource from multiple MNOs. Measurement study shows that significant differences exist in service quality in virtual networks depending on the physical network provider, application type, and location [5]. Therefore, in such an application scenario, the virtual operator (e.g., Google Fi) needs to learn the service availability and quality of different physical service providers, depending on the location, time, and application requirements of its users. At the same time, it needs to allocate its user traffic to the physical networks based on available information, subject to the service quality requirements. In this paper, we study scheduling algorithms without prior knowledge to maximize the profit of MNVOs subject to QoS constraints.

Plenty of work has been done on wireless scheduling in literature [6–11]. In particular, Lyapunov optimization approaches, *drift-plus-penalty*, are proposed to asymptotically optimize temporally average performance subject to constraints [6]. Recent work [8] studies Lyapunov optimization approaches for maximizing profit of cognitive MVNO, which can serve its subscribers by either leasing spectrum resource from MNOs or accessing the spectrum as a secondary users. Although these algorithms do not require the knowledge about the distribution of system states, they assume the reward or cost of an action under any state is known. In practice, however, an MVNO may not be able to obtain the actual value of the reward or service quality. For instance, the reward of MVNO depends on the amount of MVNO customers and their MNO traffic. The statics of reward are unknown if the traffic statics is unknown, which typically occurs at the initial stage of a time-varying system. Also it is difficult for an MVNO to know all the actual QoS of all MNOs when deploying its virtual networks. Scheduling algorithms under systems with unknown statistics are studied in wireless networks. [9] proposes a Max-Weight learning algorithm to minimize the cost in unknown environments, where the system state can only be partially revealed. However, the reward and cost are *known* functions of the system state. [10] proposes a learning plus scheduling algorithm to achieve the throughput optimality under unknown channel/estimator statistics, but it only considers the stability of the system rather than the reward optimization in our paper. The most related work [11] considers reward optimization problems with unknown reward or state distributions, but with known arrival and departure rate under a given action. In contrast, the QoS of different operators, which is used to update the virtual queue and equivalent in the arrival/departure rate in [11], need to be estimated. This makes the analysis of the system more challenging.

In this paper, we study scheduling algorithms for MVNO without any prioir information on the statistics of the reward, service quality, or system state. We propose a Learning-aided Scheduling (LSchd) algorithm for the case where the MVNO makes decision while learning the information of reward and service quality. The main idea is to use the Lyapunov optimization approach by constructing virtual queues for the

QoS requirements, and make decisions based on the estimated information of reward and QoS. We can show that as long as the estimates of the reward and QoS is close enough to the true value, the LSchd achieve a $O(1/V)$-optimal performance for a large parameter V. Further, we propose a Dual learning-aided Scheduling (DSchd) algorithm to accelerate the convergence speed by exploiting the obtained information of system states. We evaluate the proposed LSchd and DSchd algorithms by running simulations based on data collected from practical networks. The results show that both LSchd and DSchd achieve near optimal-performance while the DSchd algorithm converge faster, which is very useful in practical systems. Even when the system is non-stationary, the proposed systems can guarantee desired QoS of different applications.

The key technical contribution is to apply Lyapunov optimization method in MVNO traffic scheduling. We consider the most practical case where the reward and cost are both unknown. We leverage and integrate the existing techniques [10] [11] for Lyapunov drift analysis with inaccurate reward and cost, and use trace-driven approach to evaluate the QoS performance. The remaining of the paper is organized as follows. Section II describes the MVNO model. Section III proposes the learning aided scheduling algorithms, LSchd and DSchd, and Section IV analyzes their performance theoretically. Section V presents the simulation results and Section VI concludes our work.

II. System Model

We consider an MVNO system shown in Fig. 1, where the MVNO provides wireless communication services to its customers by utilizing network resource from multiple MNOs or WiFi networks. The system is operated in a slotted manner, i.e., the time $t \in \{0, 1, 2, \dots\}$. We note that the MVNO controls the network in an abstracted level and operates in a large time-scale. Hence, we assume the slot-length is on the order of seconds. We focus on the downlink of the MVNO system, and our techniques can be extended to the uplink by collecting the mobile-side information such as queue lengths.

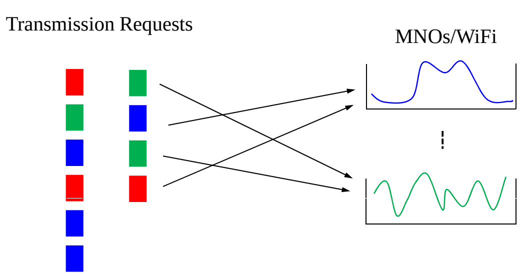


Fig. 1. Dynamic scheduling for MVNO. Transmission requests from different applications are assigned to different MNOs based on service quality.

A. Demand Model

We consider a general case where the MVNO supports multiple types of applications. Let K be the number of application types, and $A_k(t)$ be the number of type-k users arriving at time t. We assume that the arrival rate is bounded as $0 \le \sum_{k=1}^{K} A_k(t) \le A_{\max}$. Let $\mathcal{A}(t)$ be the set of users arriving at time t, and $\mathcal{A}_k(t) \subseteq \mathcal{A}(t)$ be the set of type-k users. We assume that each user arriving at time t should be either served or rejected, which provides service similar to current cellular providers. For each type k, $A_k(t)$'s are i.i.d. random variables with the expectation $\mathbb{E}[A_k(t)] = \lambda_k$. However, prior knowledge about the distribution or the expectation of $A_k(t)$ is unknown to the MVNO.

B. MNO Model

The MVNO serves its customers by acquring network resource from existing MNOs or WiFi. Note that a cellular BS and a WiFi AP have no conceptual differences, except that the service quality and the price may have different characteristics. Thus, we treat WiFi as a special MNO. Let N be the number of MNOs that carry transmissions for the MVNO. The available resource provided by the n-th MNO for the MVNO, denoted as $B_n(t)$, is time varying due to the variations of its own traffic.

1) Pricing Model: The MVNO charges its customers and pays the MNOs based on the amount of transmitted data. Let $g_n(t)$ be the per-unit-size profit, i.e., the difference between the price charged from the user and that paid to the MNO, of transmitting a unit-size of data (e.g., per gigabytes) through the n-th MNO. Moreover, an MNO can also apply dynamic pricing to coordinate the transmission. We let $C_n(t)$ be the additional price charged by the n-th MNO at time t for each user, and let $R_n(t)$ be the actual amount of transmitted data by the n-th application. Based on the pricing model, we have that the total utility at time t is

$$U(t) = \sum_{i \in \mathcal{A}(t)} [g_{n(i)} R_i(t) - C_{n(i)}(t)]. \tag{1}$$

2) QoS Model: The service quality and amount of data transmitted by user i is random due to the network conditions. Let $Y_i(t)$ be the service quality experienced by user i, which follows an unknown distribution. Let $k(i)$ be the application type of user i, and $n(i)$ be the MNO allocated to serve i. The distribution of $Y_i(t)$ depends on the application type $k(i)$ and the serving MNO $n(i)$, and its cumulative probability function is given as

$$\varphi_{k,n}(y) = \mathbb{P}\{Y_i(t) \le y | k(i) = k, n(i) = n\}.$$

The user of type-k application is unsatisfied when the quality is worse than y_k, whose probability is given as follows when it is served by the n-th MNO:

$$p_{k,n} = \varphi_{k,n}(y_k) = \mathbb{P}\{Y_i(t) \le y_k | k(i) = k, n(i) = n\}.$$

The number of users with unsatisfied service quality is denoted as

$$Z_k(t) = \sum_{i \in \mathcal{A}_k(t)} \mathbb{1}(Y_i(t) \le y_k). \tag{2}$$

C. Online Matching and Utility Maximization

Let $X(t) = (\boldsymbol{A}(t), \boldsymbol{B}(t), \boldsymbol{C}(t))$ be the system state at time t, where $\boldsymbol{A}(t) = (A_1(t), A_2(t), \ldots, A_K(t))$, $\boldsymbol{B}(t) = (B_1(t), B_2(t), \ldots, B_N(t))$, and $\boldsymbol{C}(t) = (C_1(t), C_2(t), \ldots, C_N(t))$ are the arrival state, available resouce, and price, repectively. We assume that $X(t)$ is i.i.d. and is from a finite set, i.e., $X(t) \in \mathcal{X} = \{x^{(1)}, x^{(2)}, \ldots, x^{(J)}\}$. Let $\pi_j = \mathbb{P}\{X(t) = x^{(j)}\}$ for $j = 1, 2, \ldots, J$.

At each time slot, the MVNO serves its users by leasing network resources, such as bandwidth, from MNOs. Specifically, the MVNO maps each user $i \in \mathcal{A}(t)$ to one of the MNOs based on the system state $X(t)$ and the historic information. The amount of resource required by each user is fixed and determined by its application type and the serving MNO. Let $m_{k,n}$ be the amount of resource required by type-k application at MNO n. Hence, in each time slot, the feasible mapping satisfies the following resource constraint:

$$\sum_{i \in \mathcal{A}(t): n(i) = n} m_{k(i),n} \leq B_n(t). \tag{3}$$

Each type of application has an average QoS requirement represented by (y_k, η_k), i.e., $\mathbb{P}\{Y_i(t) \leq y_k | k(i) = k\} \leq \eta_k$. On the other hand, $R_i(t)$ follows certain unknown distribution with expectation $\mathbb{E}[R_i(t)|k(i) = k, n(i) = n] = r_{k,n}$.

We consider the temporally average performance, and defined the average utility and quality violation as follows:

$$\bar{U} = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[U(t)], \tag{4}$$

$$\bar{Z}_k = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[Z_k(t)] \tag{5}$$

The objective of the MVNO is to maximize the average utility subject to the QoS requirement:

$$\max \quad \bar{U} \tag{6}$$

$$\text{s. t.} \quad \bar{Z}_k / \lambda_k \leq \eta_k, \forall k. \tag{7}$$

If the MVNO knows the system statistics, it can solve the above problem by linear programming. However, we consider a cold start problem where the system statistics, including the arrival rates, the serving rates and service quality, are unknown. We also do not assume the knowledge of the violation probability, which is different from the previous work [11]. In this paper, we discuss how to solve this constrained problem by learning-aided scheduling.

III. Learning-Aided Scheduling

We use the Lyapunov stochastic optimization techniques [6] to solve the network utility maximization problem with QoS constraints. We first present Backpressure [12] when we assume the distributions of amount of transmitted traffic and service quality are known. We then propose LSchd where we do not assume prior knowledge on the previous system statistics. Last, we propose DSchd that has the same assumption as LSchd but utilizes the learned system state information and dual learning to speed up the convergence rate.

A. Backpressure

When the expected amount of transmitted data $r_{k,n}$ and the QoS violation probability $p_{k,n}$ are known, we can solve the constrained utility maximization problem with the *drift-plus-penalty* method.

Specifically, note that the QoS constraint (7) can be rewriten as

$$\bar{Z}_k \leq \eta_k \lambda_k = \eta_k \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} A_k(t). \tag{8}$$

Thus, we introduce a counter ("virtual queue") $\Theta_k(t)$, such that $\Theta_k(0) = 0$, and

$$\Theta_k(t+1) = \Theta_k(t) - \eta_k A_k(t) + Z_k(t). \tag{9}$$

Note that we do not require $\Theta_k(t)$ to be nonnegative as existing work [6, 8–12]. This will not affect the convergence of the proposed algorithms, as will be discussed later. Let $\boldsymbol{\Theta}(t) = (\Theta_1(t), \Theta_2(t), \ldots, \Theta_K(t))$ and define the Lyapunov function as $L(t) = \frac{1}{2}||\boldsymbol{\Theta}(t)||^2$, where $||\cdot||$ is the ℓ_2-norm. For a given parameter $V > 0$, we define the one-slot utility-based conditional Lyapunov drift as follows:

$$\Delta_V(t) = \mathbb{E}\Big\{L(t+1) - L(t) - VU(t) \big| \boldsymbol{\Theta}(t)\Big\}. \tag{10}$$

According to the evolution of $\boldsymbol{\Theta}(t)$, we can show that

$$\begin{aligned} &\Delta_V(t) \\ &\leq K - \mathbb{E}\Big\{VU(t) - \sum_{k=1}^{K} \Theta_k(t)(Z_k(t) - \eta_k A_k(t)) \big| \boldsymbol{\Theta}(t)\Big\} \\ &\leq K - \mathbb{E}\Big\{\sum_{i \in \mathcal{A}(t)} w_{i,n(i)}(t) + \sum_{k=1}^{K} \eta_k A_k(t) \Theta_k(t) \big| \boldsymbol{\Theta}(t)\Big\}, \end{aligned} \tag{11}$$

where $w_{i,n}(t)$ is given by

$$w_{i,n}(t) = V\big[g_n r_{k(i),n} - C_n(t)\big] - p_{k(i),n} \Theta_k(t).$$

Then, the MVNO obtains the scheduling decision to minimize the Lyapunov drift $\Delta_V(t)$ by solving the following problem:

$$\begin{aligned} (\mathcal{P}1) \max_{n(i)} \quad & \sum_{i \in \mathcal{A}(t)} w_{i,n(i)}(t), \\ \text{s.t.} \quad & \sum_{i \in \mathcal{A}(t): n(i) = n} m_{k(i),n} \leq B_n(t). \end{aligned}$$

This is a generalized assignment problem with reward weights $w_{i,n}(t)$ and cost weights $m_{i,n}$, which is NP-hard. We assume that one user can be served by multiple MNOs and the above problem becomes a linear programming problem. This relaxation will obtain an approximate solution when there are many users.

B. LSchd

When the expected amount of data and service quality are unknown, we propose to apply estimates obtained from learning module for implementing the Lyapunov approach proposed in the previous section. We first define the capability of a learning module. For any matrix $\boldsymbol{W}$ and its estimate $\hat{\boldsymbol{W}}$ provided by a learning module, we denote the maximum estimation error as $\delta_w = ||\hat{\boldsymbol{W}} - \boldsymbol{W}||_{\max}$, where $||x||_{\max} = \max_{ij} |x_{ij}|$. We adopt the definition in [11] to capture the capability of a learning module.

Definition 1 *An algorithm Γ is called a $(T_\delta, P_\delta, \delta)$-learning module, if (i) it completes learning in T_δ slots, (ii) it guarantees $\mathbb{P}\{\delta_w < \delta\} \geq P_\delta$, and (iii) P_δ does not decrease if the algorithm is run for $T > T_\delta$ time slots.*

As discussed in [11], two sampling-based learning algorithms, Threshold-Based Sampling (TBS) and Time-Limited Sampling (TLS), are examples of this type of learning module.

We propose a Learning-aided Scheduling (LSchd) that applies the estimates of the transmitting rate and QoS for scheduling for MVNO, as shown in Algorithm 1.

Algorithm 1 Learning-aided Scheduling (LSchd)

Learning: Apply any $(T_{\delta_r}, P_{\delta_r}, \delta_r)$-learning module for $\boldsymbol{r}$ and any $(T_{\delta_p}, P_{\delta_p}, \delta_p)$-learning module for $\boldsymbol{p}$; terminate at $T_1 = \max\{T_{\delta_r}, T_{\delta_p}\}$ and output $\hat{\boldsymbol{r}}$ and $\hat{\boldsymbol{p}}$;
Scheduling: Set $\boldsymbol{\Theta}(T_1 + 1) = \boldsymbol{0}$ and implement the drift-plus-penalty method with the estimates $\hat{\boldsymbol{r}}$ and $\hat{\boldsymbol{p}}$.

C. DSchd

In this section, we propose a scheduling algorithm that utilizes the learned system state information to speed up the convergence rate. When the system statistics are fully known, we can maximize the network utility using an $\boldsymbol{X}$-only policy that makes stationary decisions and hence independent of queue size [6]. Specifically, let $\boldsymbol{n}_j^l$ be one of a feasible scheduling when the system state is $x^{(j)}$ and an $\boldsymbol{X}$-only policy $\boldsymbol{\alpha}$ defines the probability of choosing one of the feasible schedules under any state. The MVNO obtains the optimal $\boldsymbol{X}$-only policy by solving the following problem.

$$
\begin{aligned}
(\mathcal{P}2)\ \max_{\boldsymbol{\alpha}}\ & V \sum_{j=1}^{J} \pi_j r_j(\boldsymbol{\alpha}), \\
\text{s.t. } & \sum_{j=1}^{J} \pi_j z_{j,k}(\boldsymbol{\alpha}) \leq \eta_k \lambda_k, \forall k, \\
& \boldsymbol{n}_j \in \mathcal{N}^{(j)}, \forall j.
\end{aligned}
$$

where $r_j(\boldsymbol{\alpha})$ is the expected reward and $z_{j,k}(\boldsymbol{\alpha})$ is the expected number of type-k QoS violations at state x_j under policy $\boldsymbol{\alpha}$. This problem is an LP problem.

On the other hand, the Lagrangian is given by

$$
\mathcal{L}(\boldsymbol{\gamma}, \boldsymbol{\alpha}) = V \sum_{j=1}^{J} \pi_j r_j(\boldsymbol{\alpha}) - \sum_{k=1}^{K} \gamma_k \big[\sum_{j=1}^{J} \pi_j z_{j,k}(\boldsymbol{\alpha}) - \eta_k \lambda_k \big], \tag{12}
$$

Where $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_K)$ is the Lagrangian multiplier.
The dual function is defined as

$$
G(\boldsymbol{\gamma}) = \max_{\boldsymbol{\alpha}} \mathcal{L}(\boldsymbol{\gamma}, \boldsymbol{\alpha}). \tag{13}
$$

The dual problem is

$$
\begin{aligned}
& \min\ G(\boldsymbol{\gamma}), \\
& \text{s.t. } \boldsymbol{\gamma} \succeq \boldsymbol{0}.
\end{aligned} \tag{14}
$$

When the system statistics are known, we can obtain the Lagrangian multipliers by solving the above dual problem. Otherwise, we can obtain the approximate multipliers by solving the empirical dual problem. In this case, we propose a Dual Learning-aided Scheduling algorithm. As shown in Algorithm 2, we first learn the system statistics, and then use the estimates to solve the dual problem. After that, we choose the scheduling decisions by using the estimate Lagrangian multipliers as the initial value of the "counters". Specifically, we introduce another counter $\tilde{\Theta}_k(t)$, such that $\tilde{\Theta}_k(0) = 0$, and

$$
\tilde{\Theta}_k(t) = \Theta_k(t) + \hat{\gamma}_k^* - \zeta_k \tag{15}
$$

where $\zeta_k = (\log V)^2$. Also we have

$$
\Theta_k(t) = \max(\zeta_k - \hat{\gamma}_k^*, \Theta_k(t-1) - \eta_k A_k(t)) + Z_k(t) \tag{16}
$$

Algorithm 2 Dual Learning-aided Scheduling (DSchd)

Learning:
(1) Apply any $(T_{\delta_r}, P_{\delta_r}, \delta_r)$-learning module for $\boldsymbol{r}$, $(T_{\delta_p}, P_{\delta_p}, \delta_p)$-learning module for $\boldsymbol{p}$, and $(T_{\delta_\pi}, P_{\delta_\pi}, \delta_\pi)$-learning module for $\boldsymbol{\pi}$; terminate at $T_2 = \max\{T_{\delta_r}, T_{\delta_p}, T_{\delta_\pi}\}$ and output $\hat{\boldsymbol{r}}$, $\hat{\boldsymbol{p}}$, and $\hat{\boldsymbol{\pi}}$;
(2) Solve the empirical dual problem with $\hat{\boldsymbol{r}}$, $\hat{\boldsymbol{p}}$, and $\hat{\boldsymbol{\pi}}$.

$$
\begin{aligned}
& \min\ G(\boldsymbol{\gamma}), \\
& \text{s.t. } \boldsymbol{\gamma} \succeq \boldsymbol{0}.
\end{aligned} \tag{17}
$$

Let $\hat{\boldsymbol{\gamma}}^* = (\hat{\gamma}_1^*, \hat{\gamma}_2^*, \ldots, \hat{\gamma}_K^*)$ be the optimal solution.
Scheduling: Set $\boldsymbol{\Theta}(T_2+1) = \boldsymbol{0}$; Assign the users by solving the assignment problem $\mathcal{P}$ with $\hat{\boldsymbol{r}}$, $\hat{\boldsymbol{p}}$, $\hat{\boldsymbol{\pi}}$, and $\tilde{\boldsymbol{\Theta}}(t)$, where $\Theta_k(t) = \max(\zeta_k - \hat{\gamma}_k^*, \Theta_k(t-1) - \eta_k A_k(t)) + Z_k(t)$ and $\tilde{\Theta}_k(t) = \Theta_k(t) + \hat{\gamma}_k^* - \zeta_k$ with $\zeta_k = (\log V)^2$.

IV. Performance Analysis

In this section, we study the performance of the proposed LSchd and DSchd algorithms. To indicate the different information used, we use $\hat{G}(\boldsymbol{\gamma})$ to denote the dual function when $\boldsymbol{r}$ is replaced with $\hat{\boldsymbol{r}}$ and $\boldsymbol{p}$ is replaced with $\hat{\boldsymbol{p}}$, and the actual distribution $\boldsymbol{\pi}$ is used. Similarly, we use $G^{\hat{\boldsymbol{\pi}}}(\boldsymbol{\gamma})$ and $\hat{G}^{\hat{\boldsymbol{\pi}}}(\boldsymbol{\gamma})$ to denote the dual function with $(\boldsymbol{r}, \boldsymbol{p}, \hat{\boldsymbol{\pi}})$ and $(\hat{\boldsymbol{r}}, \hat{\boldsymbol{p}}, \hat{\boldsymbol{\pi}})$, respectively.

A. Preliminaries

We define the following polyhedral system structure:

Definition 2 *A system is polyhedral with parameter $\rho > 0$ if the dual function $G(\boldsymbol{\gamma})$ satisfies:*

$$G(\boldsymbol{\gamma}^*) \leq G(\boldsymbol{\gamma}) - \rho||\boldsymbol{\gamma}^* - \boldsymbol{\gamma}||, \tag{18}$$

where $\boldsymbol{\gamma}^*$ is the optimal dual solution.

We also make the following assumptions.

Assumption 1 *There exist constants $\epsilon_r, \epsilon_p, \epsilon_\pi = \Theta(1) > 0$ such that for any valid estimates $\hat{\boldsymbol{r}}$, $\hat{\boldsymbol{p}}$, and $\hat{\boldsymbol{\pi}}$ with $||\hat{\boldsymbol{r}} - \boldsymbol{r}|| \leq \epsilon_r$, $||\hat{\boldsymbol{p}} - \boldsymbol{p}|| \leq \epsilon_p$, and $||\hat{\boldsymbol{\pi}} - \boldsymbol{\pi}|| \leq \epsilon_\pi$, there exits a set of actions $(\boldsymbol{n}_j^l : 1 \leq j \leq J, 1 \leq l \leq \infty)$ and variables$(\beta_j^l : 1 \leq j \leq J, 1 \leq l \leq \infty)$ with $\sum_l \beta_j^l = 1$ (possibly depending on the estimates), such that*

$$\sum_{j=1}^{J} \hat{\pi}_j \sum_l \beta_j^l [\sum_{i \in \mathcal{A}^{(j)}} \hat{p}_{k(i), n_j^l(i)} - \eta_k \lambda_k] \leq -\eta_0 \lambda_k, \tag{19}$$

where $\eta_0 = \Theta(1) > 0$ is independent of $\hat{\boldsymbol{r}}$, $\hat{\boldsymbol{p}}$, and $\hat{\boldsymbol{\pi}}$

Assumption 2 *There exist constants $\epsilon_r, \epsilon_p, \epsilon_\pi = \Theta(1) > 0$ such that for any valid estimates $\hat{\boldsymbol{r}}$, $\hat{\boldsymbol{p}}$, and $\hat{\boldsymbol{\pi}}$ with $||\hat{\boldsymbol{r}} - \boldsymbol{r}|| \leq \epsilon_r$, $||\hat{\boldsymbol{p}} - \boldsymbol{p}|| \leq \epsilon_p$, and $||\hat{\boldsymbol{\pi}} - \boldsymbol{\pi}|| \leq \epsilon_\pi$, if $G(\boldsymbol{\gamma})$ is polyhedral with parameter ρ, then $\hat{G}^{\boldsymbol{\pi}}(\boldsymbol{\gamma})$ is also polyhedral with parameter ρ.*

Assumption 3 *There exist constants $\epsilon_r, \epsilon_p, \epsilon_\pi = \Theta(1) > 0$ such that for any valid estimates $\hat{\boldsymbol{r}}$, $\hat{\boldsymbol{p}}$, and $\hat{\boldsymbol{\pi}}$ with $||\hat{\boldsymbol{r}} - \boldsymbol{r}|| \leq \epsilon_r$, $||\hat{\boldsymbol{p}} - \boldsymbol{p}|| \leq \epsilon_p$, and $||\hat{\boldsymbol{\pi}} - \boldsymbol{\pi}|| \leq \epsilon_\pi$, $\hat{G}^{\boldsymbol{\pi}}(\boldsymbol{\gamma})$ space has a unique optimal solution.*

Let $\Psi(\boldsymbol{\eta})$ be the optimal value of problem $\mathcal{P}2$, using the perturbation-analysis for LP problems [13] we have the following lemma.

Lemma 1 *For a given set of parameters satisfying Assumption 1, for any $\epsilon_0 \in [0, \eta_0]$ and $\boldsymbol{\eta} - \epsilon_0 \mathbf{1}$, there is an $\boldsymbol{X}$-only policy satisfying*

$$\sum_{j=1}^{J} \pi_j z_{j,k}(\boldsymbol{\alpha}) \leq (\eta_k - \epsilon_0)\lambda_k,$$
$$V \sum_{j=1}^{J} \pi_j r_j(\boldsymbol{\alpha}) = \Psi(\boldsymbol{\eta} - \epsilon_0 \mathbf{1}).$$

B. Utility and Constraint Performance

We first present the utility performance for LSchd.

Theorem 1 *Suppose $T_1 = \max(T_{\delta_r}, T_{\delta_p}) < \infty$ with probability 1. Under LSchd, we have with probability $P_{\delta_r} P_{\delta_p}$ that*

$$\bar{U}^{\text{LSchd}} \geq \bar{U}^* - \frac{K + O(\delta_r + \delta_p)}{V} - 2\delta_r A_{\max} g_{\max}. \tag{20}$$

and

$$\bar{Z}_k^{\text{LSchd}} \leq \eta_k \lambda_k, \ \forall k. \tag{21}$$

Note that there is performance loss due to estimation error δ_r in the last term, while we could guarantee the desired QoS for all types of applications. Then we present the performance result for DSchd.

Theorem 2 *Suppose $T_2 = \max(T_{\delta_r}, T_{\delta_p}, T_{\delta_\pi}) < \infty$ with probability 1. Under DSchd, we have with probability $P_{\delta_r} P_{\delta_p} P_{\delta_\pi}$ that*

$$\bar{U}^{\text{DSchd}} \geq \bar{U}^* - \frac{K + O(\delta_r + \delta_p)}{V} - 2\delta_r A_{\max} g_{\max} - O(\frac{1}{V}). \tag{22}$$

and

$$\bar{Z}_k^{\text{DSchd}} \leq \eta_k \gamma_k, \ \forall k. \tag{23}$$

In both LSchd and DSchd, the effect of inaccurate estimation is not guaranteed to be always eliminated when increasing the learning period, since the estimation error and its evolution is largely related to the strategies made on early stage, which is sensitive to change based on system statistics.

C. Convergence Time

Convergence time plays an important role when system statistics can change. We introduce the formal definition of convergence time as follows.

Definition 3 *For a given constant D, the D-convergence time of a scheduling algorithm Π, denoted by T_D^Π, is the time it takes for the queue vector $\boldsymbol{\Theta}(t)$ to get to within D distance of $\boldsymbol{\gamma}^*$, i.e.,*

$$T_D^\Pi = \inf\{t : ||\boldsymbol{\Theta}(t) - \boldsymbol{\gamma}^*|| \leq D\}. \tag{24}$$

With definition above, we present the following results:

Theorem 3 *Suppose $G(\boldsymbol{\gamma})$ is polyhedral with $\rho = \Theta(1) > 1$, $\delta_r \leq \epsilon_r$, $\delta_p \leq \epsilon_p$, and $\delta_\pi \leq \epsilon_\pi$. Then with a sufficiently large V, we have*

$$\mathbb{E}[T_{D_1}^{\text{LSchd}}] = O(T_1 + O(V)) \ w.p. \ P_{\delta_r} P_{\delta_p},$$
$$\mathbb{E}[T_{D_2}^{\text{DSchd}}] = O(T_2 + O(\delta_p V)) \ w.p. \ P_{\delta_r} P_{\delta_p}.$$

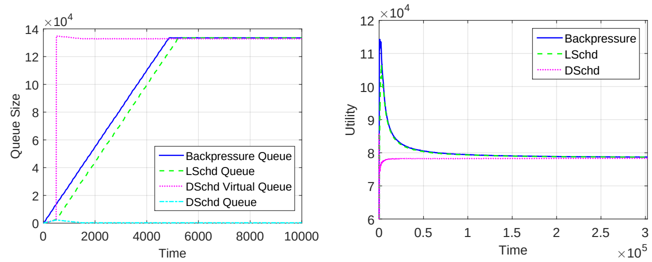
V. SIMULATION

We now provide the simulation results for LSchd and DSchd to demonstrate the near-optimal performance and fast convergence speed. The proposed algorithms are evaluated based on traces collected from real cellular network, provided by Speedometer[1] and Mobile Network Dataset. Speedometer is an Android custom mobile network measurement app developed by Google and is running on thousands of volunteer phones. The data consists of ping, traceroute, DNS lookup, HTTP fetches, and UDP packet-loss measurements from 2011-10 to 2013-08. Mobile Network Dataset comes from a tier-1 operator in China, which records all the session details of 5 neighboring base stations for one week.

We consider $\boldsymbol{A}(t)$ to be the number of sessions at time t, and $\boldsymbol{R}(t)$ to be the total downlink size of sessions at time t, both from Mobile Network Dataset. We also consider the reciprocal of average RTT of Verizon, T-Mobile and Sprint from Speedometer dataset as a metric of user experienced

[1]This dataset is available at https://storage.cloud.google.com/speedometer

service quality $\boldsymbol{Y}(t)$. We consider a system that has $K = 2$, $N = 3$, $A = 5$ and $B = 2$, where A and B are quantization levels of arrivals and available resources. In this case we have a total of 200 different system state combinations. We also have per-unit-size profit $g_n(t) = \{0.4, 0.6, 0.2\}$, quality requirement level $\boldsymbol{y} = \{0.2, 0.25\}$, and quality requirement probability η to be 5% for both application types.



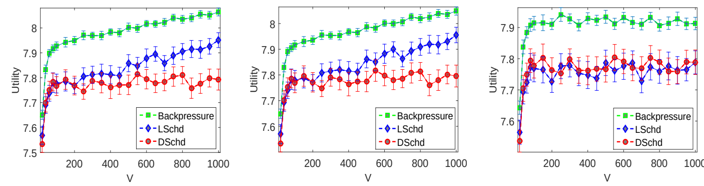
(a) Convergence performance (b) Utility performance

Fig. 2. Performance under three algorithms

We compare the proposed two algorithms LSchd and DSchd with Backpressure, which assumes the distributions of amount of transimitted traffic and service quality are known. Figs. 2(a) and 2(b) show that LSchd and DSchd achieve near-optimal performance even when the system statistics are non-stationary. In Fig. 2(a), we can see the behavior of the first queue under the three different algorithms with $V = 300$. The average queue size after being stable is indistinguishable among Backpressure, LShed and DSchd, which indicates same scheduling decisions in the long run. In Fig. 2(b), LSchd and DSchd could also achieve near-optimal average utility when compared with Backpressure. For Backpressure and LSchd, the queue size is not accumulated in the early slots, therefore the decision is inclined to high-profit-low-quality MNOs which dramatically increase the utility. While for DSchd, the queue size is quickly raised to the stable level after the short learning period, thus the average utility changes smoothly.

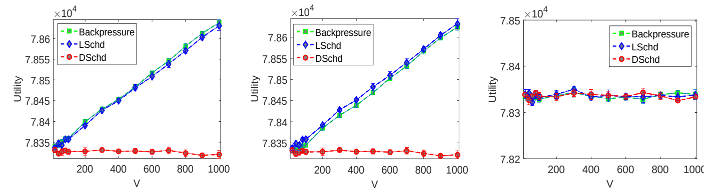
Then we look at the convergence rate of the algorithms. As shown in Fig. 2(a), the first virtual queue and actual queue size under DSchd converge much faster than Backpressure and LSchd, which is very useful in practical systems. For example, when $V = 300$, the convergence time of Backpressure is 4900 timeslots, while DSchd is only about 1500 timeslots. The performance of the second queue is similar and we do not present here. The quick convergence rate demonstrates the power of learning-aided scheduling techniques, especially when the system statistics are non-stationary as in our case.

Figs. 3 to 6 show the behavior of the proposed algorithms when we tune V and the length of learning period. For comparison, besides the real trace scenario, we also simulate using synthetic data. We consider a system with $K = 2$, $N = 3$, $A = 2$ and $B = 2$. We have $g_n(t) = \{0.6, 0.4, 0.2\}$, $\boldsymbol{y} = \{0.1, 0.2\}$, and η to be 4.5% and 9% respectively, in order to simulate applications with different quality-violation tolerance. We assume that $\boldsymbol{A_j}(t)$ is i.i.d with either 10 or



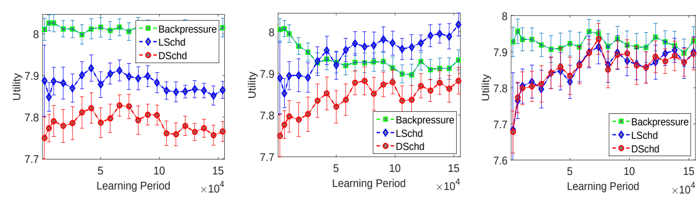
(a) Overall (b) After Learning (c) Stable

Fig. 3. Average utility under sythetic data when changing V



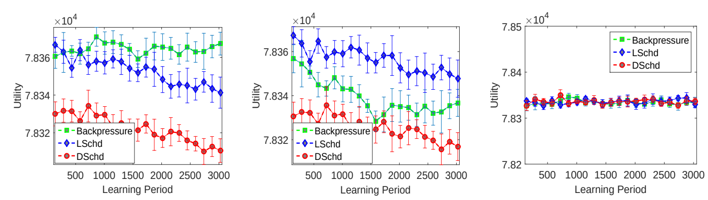
(a) Overall (b) After Learning (c) Stable

Fig. 4. Average utility under real trace when changing V



(a) Overall (b) After Learning (c) Stable

Fig. 5. Average utility under synthetic data when changing learning period



(a) Overall (b) After Learning (c) Stable

Fig. 6. Average utility under real trace when changing learning period

20 with probabilities p_j and $1 - p_j$, where $p_1 = 0.3$ and $p_2 = 0.4$. The mean value of service quality experienced by user is $\boldsymbol{Y_{mean}} = \{1, 1.2, 1.4\}$, and the mean amount of transmitted data is $\boldsymbol{R_{mean}} = \{0.5, 0.5\}$, both of which are uniformly distributed. For better illustration, we separately plot the average utility in the whole period, the after-learning period and the stable period, where the after-learning period starts from the end of the learning period, and the stable period starts when the system statistics are stabilized.

Firstly we see from Figs. 3 and 4 that the overall average utility increases as we increase the value of V, especially in Fig. 3(c) which indicates a $O(1/V)$ close-to-optimal performance. The utility gap between Backpressure and learning-aided algorithms in the stable stage of synthetic data is due to the insufficient learning length, which is solved in real trace scenario when we have long enough learning slots. Notice that LSchd and DSchd have different sentitivity towards V. Also notice that even for Backpressure, when V is large enough, the performance is hard to improve further.

Figs. 5 and 6 then show the utility performance with different learning length. We see that in the after-learning stage, as learning period increases, LSchd and DSchd achieve higher average uility since the system dynamics is learned more accurately. All the algorithms converge to the same value in the stable stage under a sufficient learning period. We also observe that when learning period is overly large, the average utility begins to drop. This is because the algorithm has learned fairly well already, while wasting a lot performance to explore different strategies in learning stage. Since the real trace data is periodic, it's easier to learn correctly in a short period, thus the utility is always dropping.

VI. Conclusion

In this paper, we study the scheduling problem with QoS constraints for MVNOs without *a priori* knowledge. We propose two learning-aided scheduling algorithms LSchd and DSchd for the case where the MVNO makes decision by leveraging the information of reward and service quality based on Lyapunov optimization approaches. We show that even when the system statistics are non-stationary, both LSchd and DSchd can achieve a $O(1/V)$-optimal network utility subject to average QoS constraints for a large parameter V, and DSchd significantly improves the convergence speed. We evaluate the proposed LSchd and DSchd algorithms by simulations based on traces collected from real cellular network, which guarantee the QoS of different application and provide insights into the design of learning-aided algorithms for practical systems.

Acknowledgments: The work was partially supported by NSF (Grants: CNS-1547461, CNS-1457060, CCF-1423542), NBRPC (Grants: 2011CBA00300, 2011CBA00301) and NSFC (Grants: 61361136003, 61303195).

References

[1] C. Liang and F. R. Yu, "Wireless virtualization for next generation mobile cellular networks," *Wireless Communications, IEEE*, vol. 22, no. 1, pp. 61–69, 2015.

[2] "Project Fi," https://fi.google.com/about/.

[3] "Skytone," https://skytone.vmall.com/.

[4] Grand View Research, "Mobile Virtual Network Operator (MVNO) market analysis and segment forecasts to 2020," Tech. Rep., 2015.

[5] F. Zarinni, A. Chakraborty, V. Sekar, S. R. Das, and P. Gill, "A first look at performance in mobile virtual network operators," in *Proceedings of the 2014 Conference on Internet Measurement Conference*. ACM, 2014, pp. 165–172.

[6] 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.

[7] H. Wu, X. Lin, X. Liu, K. Tan, and Y. Zhang, "CoSchd: coordinated scheduling with channel and load awareness for alleviating cellular congestion," *IEEE/ACM Trans. on Networking*, to appear.

[8] S. Li, J. Huang, and S.-Y. R. Li, "Dynamic profit maximization of cognitive mobile virtual network operator," vol. 13, no. 3, pp. 526–540, 2014.

[9] M. J. Neely, S. T. Rager, and T. F. La Porta, "Max weight learning algorithms for scheduling in unknown environments," *IEEE Trans. on Automatic Control*, vol. 57, no. 5, pp. 1179–1191, 2012.

[10] W. Ouyang, S. Murugesan, A. Eryilmaz, and N. B. Shroff, "Scheduling with rate adaptation under incomplete knowledge of channel/estimator statistics," in *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*. IEEE, 2010, pp. 670–677.

[11] L. Huang, "The value-of-information in matching with queues," in *ACM MobiHoc*. ACM, 2015.

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

[13] J. Renegar, "Some perturbation theory for linear programming," *Mathematical Programming*, vol. 65, no. 1, pp. 73–91, 1994.

[14] L. Huang, X. Liu, and X. Hao, "The power of online learning in stochastic network optimization," in *ACM Sigmetrics*. ACM, 2014, pp. 153–165.

Appendix A
Proof of Theorem 1

Let

$$\Phi_t(\boldsymbol{n}) = \sum_{i\in\mathcal{A}(t)} \big\{V[g_{n(i)}r_{k(i),n(i)} - C_{n(i)}(t)] - p_{k(i),n(i)}\Theta_{k(i)}(t)\big\}.$$

Lemma 2 *Under LSched, we have with probability* $P_{\delta_r}P_{\delta_p}$ *that for each time* $t > T_1$,

$$|\Phi_t(\boldsymbol{n}_t) - \Phi_t(\hat{\boldsymbol{n}}_t)| \le 2\delta_r A_{\max} g_{\max} V + 2\delta_p A_{\max}\sum_{k=1}^{K}\Theta_k(t),$$

where $\boldsymbol{n}_t$ *is the optimal scheduling using the drift-plus-penalty with full size and QoS information, and* $\hat{\boldsymbol{n}}_t$ *is the scheduling under LSched.*

The idea is to compare the utility of optimal scheduling with utility of sub-optimal scheduling under the same accurate system statistics. We could bound the difference through an intermediate step using the utility of sub-optimal scheduling under estimated system information.

Let $\boldsymbol{\alpha}$ be the optimal X-only policy for $\boldsymbol{\eta} - \epsilon_0\mathbf{1}$ for any $\epsilon_0 \in [0, \eta_0]$, then

$$\Phi_t(\boldsymbol{n}_t) \ge \Psi(\boldsymbol{\eta} - \epsilon_0\mathbf{1}) - \sum_{k=1}^{K}\Theta_k(t)(\eta_k - \epsilon_0)\lambda_k. \quad (25)$$

Then we calculate the Lyapunov drift under LSched. Taking an expectation over $\boldsymbol{\Theta}(t)$, carrying out a telescoping sum from $t = 0$ to $T-1$, we have

$$\begin{aligned}&\mathbb{E}[L(T) - L(0)] - V\sum_{t=0}^{T-1}U(t)\\ &\le KT - T\Psi(\boldsymbol{\eta} - \epsilon_0\mathbf{1}) + 2\delta_r A_{\max} g_{\max} TV\\ &\quad - (\epsilon_0\lambda_{\min} - 2\delta_p A_{\max})\sum_{t=0}^{T-1}\sum_{k=1}^{K}\Theta_k(t). \quad (26)\end{aligned}$$

Let $\epsilon_0 = 2\delta_p A_{\max}/\lambda_{\min}$. If the estimate of p satisfies $2\delta_p A_{\max}/\lambda_{\min} \le \eta_0$, we have

$$\frac{1}{T}\sum_{t=0}^{T-1}U(t) \ge \frac{1}{V}\Psi(\boldsymbol{\eta} - \epsilon_0\mathbf{1}) - \frac{K}{V} - 2\delta_r A_{\max} g_{\max}. \quad (27)$$

By perturbation theory for linear programming [13], we have

$$|\Psi(\boldsymbol{\eta}) - \Psi(\boldsymbol{\eta} - \epsilon_0 \mathbf{1})| \leq O(\delta_r + \delta_p) \tag{28}$$

Taking a limit as $T \to \infty$, we have

$$\bar{U}^{\text{LSched}} \geq \bar{U}^* - \frac{K + O(\delta_r + \delta_p)}{V} - 2\delta_r A_{\max} g_{\max}. \tag{29}$$

On the other hand, when $2\delta_p A_{\max}/\lambda_{\min} \leq \eta_0/2$ and let $\epsilon_0 = \eta_0$, we have

$$\frac{\eta_0 \lambda_{\min}}{2} \sum_{k=1}^{K} \bar{\Theta}_k \leq \frac{1}{T} \sum_{t=0}^{T-1} U(t) + K + \epsilon_1 V < \infty. \tag{30}$$

Given the stability of time-averaging virtual queues, we have $\bar{Z}_k \leq \eta_k \lambda_k$ for all k.

Appendix B
Proof of Theorem 2

Let $\tilde{\boldsymbol{\gamma}}^*$ be the optimal solution for $\hat{G}(\boldsymbol{\gamma})$, and $\hat{\boldsymbol{\gamma}}^*$ be the optimal solution for $\hat{G}^{\hat{\boldsymbol{\pi}}}(\boldsymbol{\gamma})$.

Lemma 3 *Suppose $G(\boldsymbol{\gamma})$ is polyhedral with $\rho = \Theta(1) > 0$, and that $\delta_r \leq \epsilon_r$, $\delta_p \leq \epsilon_p$, and $\delta_\pi \leq \epsilon_\pi$. Then, with probability $P_{\delta_r} P_{\delta_p} P_{\delta_\pi}$, we have*

$$||\boldsymbol{\gamma}^* - \tilde{\boldsymbol{\gamma}}^*|| \leq \frac{A_{\max} g_{\max} V \delta_r + 2V f_{\max} \delta_p / (\eta_0 \lambda_{\min})}{\rho},$$
$$||\hat{\boldsymbol{\gamma}}^* - \tilde{\boldsymbol{\gamma}}^*|| \leq \frac{A_{\max} g_{\max} V \delta_r + 2V f_{\max} \delta_p / (\eta_0 \lambda_{\min})}{\rho}. \tag{31}$$

Proof: Let $f_{\max} = A_{\max} g_{\max}$. Using Assumption 1 in Section IV and Lemma 1 in [14], we can show that with probability $P_{\delta_r} P_{\delta_p} P_{\delta_\pi}$, we have

$$\sum_{k=1}^{K} \hat{\gamma}_k \leq \frac{V f_{\max}}{\eta_0 \lambda_{\min}}, \tag{32}$$

which also holds for $\boldsymbol{\gamma}^*$ and $\tilde{\boldsymbol{\gamma}}^*$. Then similar to the analysis in Appendix of [11], we get

$$G(\boldsymbol{\gamma}^*, \boldsymbol{\alpha}^*) \geq G(\tilde{\boldsymbol{\gamma}}^*, \hat{\boldsymbol{\alpha}}^*) - 2A_{\max} g_{\max} V \delta_r - 2\frac{f_{\max} V \delta_p}{\eta_0 \lambda_{\min}}.$$

Finally we get the result by using the polyhedral of $G(\boldsymbol{\gamma})$. ■

To prove Theorem 2, we firstly need to show that given (16), $\Delta_V(t)$ should still be the same with (10). It is trivial when $\Theta_k(t) - \eta_k A_k(t) \geq \zeta_k - \hat{\gamma}_k^*$. When $\Theta_k(t) - \eta_k A_k(t) \leq \zeta_k - \hat{\gamma}_k^*$, since $\zeta_k = (\log V)^2$ and $\hat{\gamma}_k^* = O(V)$, given a sufficiently large V, $\zeta_k - \hat{\gamma}_k^*$ should be negative. Therefore we have

$$\begin{aligned}
&\Delta_V(t) \\
&\leq \mathbb{E}\bigg\{ \sum_{k=1}^{K} [\frac{1}{2}(\Theta_k(t) - \eta_k A_k(t))^2 + Z_k(t)\Theta_k(t) + \frac{1}{2} Z_k(t)^2 \\
&\quad - \frac{1}{2}\Theta_k(t)^2] - VU(t) \big| \boldsymbol{\Theta}(t) \bigg\} \\
&\leq K - \mathbb{E}\bigg\{ VU(t) - \sum_{k=1}^{K} \Theta_k(t)(Z_k(t) - \eta_k A_k(t)) \big| \boldsymbol{\Theta}(t) \bigg\}
\end{aligned}$$

Then we define the following drift-augmentation term:

$$\Delta_a(t) \triangleq \sum_{k=1}^{K} (\hat{\gamma}_k^* - \zeta) \mathbb{E}[Z_k(t) - \eta_k \lambda_k | \boldsymbol{\Theta}(t)]. \tag{33}$$

Adding it to both side of (11), similar to the proof of Theorem 1, we can show that with probability $P_{\delta_r} P_{\delta_p}$, for $\epsilon_0 \lambda_{\min} = 2\delta_p A_{\max}$ and any t,

$$\begin{aligned}
&\bar{U}^{\text{DSched}} \\
&\geq \frac{1}{V}\Psi(\boldsymbol{\eta} - \epsilon_0 \mathbf{1}) - \frac{K}{V} - 2\delta_r A_{\max} g_{\max} - \lim_{T\to\infty} \frac{1}{TV} \sum_{t=0}^{T-1} \Delta_a(t).
\end{aligned}$$

Using Theorem 2 in [11], we know that the system is stable under DSched. Thus, using Lemma 4 in [11], we have that

$$\bar{Z}_k - \eta_k \lambda_k \leq \lim_{t\to\infty} A_{\max} \mathbb{P}\{\Theta_k(t) \leq A_{\max}\} \tag{34}$$

By choosing $\nu = \frac{\ln V A_{\max}}{\kappa}$ and $D = \xi - A_{\max} - \frac{\ln V A_{\max}}{\kappa}$, we have $\bar{Z}_k - \eta_k \lambda_k \leq \xi / V$. Furthermore, we can show that $\hat{\gamma}_k = O(V)$ as in Appendix F of [11], and thus $\lim_{T\to\infty} \frac{1}{TV} \sum_{t=0}^{T-1} \Delta_a(t) = O(1/V)$.

Appendix C
Proof of Theorem 3

To prove the convergence time, we define a different Lyapunov function as follows:

$$L_0(t) = \frac{1}{2} ||\boldsymbol{\Theta}(t) - \tilde{\gamma}^*||^2, \tag{35}$$

Then, we define a one-slot conditional Lyapunov drift as $\Delta_0(t) = \mathbb{E}[L_0(t+1) - L_0(t) | \boldsymbol{\Theta}(t)]$, which can be bounded as follows

$$\Delta_0(t) \leq K + \sum_{k=1}^{K} (\Theta_k(t) - \tilde{\gamma}_k^*) \mathbb{E}[Z_k(t) - \eta_k \lambda_k | \boldsymbol{\Theta}(t)]. \tag{36}$$

Similar to the analysis of Theorem 3 in [11], we can show that the Lyapunov function has a constant negative drift. From the definition of $G(\boldsymbol{\gamma})$ and $\hat{G}(\boldsymbol{\gamma})$, we know that under LSched, $\sum_{k=1}^{K} \mathbb{E}[Z_k(t) - \eta_k \lambda_k]$ is the subgradient of $\hat{G}(\boldsymbol{\gamma})$ at $\boldsymbol{\Theta}(t)$. Thus

$$\Delta_0(t) \leq K - \rho ||\boldsymbol{\Theta}(t) - \tilde{\gamma}^*|| \tag{37}$$

When $||\boldsymbol{\Theta}(t) - \tilde{\gamma}^*|| \geq \frac{K}{\rho - \epsilon_0}$ $(0 \leq \epsilon_0 \leq \rho)$, we have

$$\mathbb{E}[||\boldsymbol{\Theta}(t+1) - \tilde{\gamma}^*|| | \boldsymbol{\Theta}(t)] \leq ||\boldsymbol{\Theta}(t) - \tilde{\gamma}^*|| - \epsilon_0 \tag{38}$$

Let $D_1' = \frac{K}{\rho - \epsilon_0}$. According to (33),

$$D_1 = D_1' + \frac{A_{\max} g_{\max} V \delta_r + 2V f_{\max} \delta_p / (\eta_0 \lambda_{\min})}{\rho} \tag{39}$$

Note that $\tilde{\gamma}_k^* = O(V)$. Thus, $\mathbb{E}[T_{D_1}^{\text{LSched}}] = \mathbb{E}[T_1 + O(V)]$.

For DSched, it uses the estimated value of Lagrangian mulitipliers. Thus, the initial value of the algorithm jumps to $\hat{\gamma}^*$. The convergence time of DSched is $\mathbb{E}[T_{D_2}^{\text{DSched}}] = \mathbb{E}[T_2 + O(\frac{2\delta_p V f_{max}}{\rho \epsilon_0 \eta})]$.