Demand Shaping in Cellular Networks

Xinyang Zhou Lijun Chen

***Abstract*— Demand shaping is a promising way to mitigate the wireless cellular capacity shortfall in the presence of ever-increasing wireless data demand. In this paper, we formulate demand shaping as an optimization problem that minimizes the variation in aggregate traffic. We design a distributed and randomized offline demand shaping algorithm under complete traffic information and prove its almost surely convergence. We further consider a more realistic setting where the traffic information is incomplete but the future traffic can be predicted to a certain degree of accuracy. We design an online demand shaping algorithm that updates the schedules of deferrable applications (DAs) each time when new information is available, based on solving at each timeslot an optimization problem over a shrinking horizon from the current time to the end of the day. We compare the performance of the online algorithm against the optimal offline algorithm, and provide numerical examples to complement the theoretical analysis.**

***Index Terms*— Demand shaping, offline algorithm, online algorithm, steepest descent algorithm, supermartingale, deferrable applications, cellular networks.**

I. Introduction

We have witnessed in recent years rapid increase in demand for wireless data, driven by the proliferation of smart mobile devices. The global mobile traffic in 2016 has nearly reached 84 exabytes, more than 80 times greater than the entire global Internet traffic in 2000; yet, this number is expected to be increasing at a compound annual growth rate (CAGR) of 47% in the coming five years, i.e., a seven-fold growth from 2016 to 2021 [20]. However, despite frequent upgrades of cellular networks technology from 2G to 4G LTE and beyond, wireless service providers fall short of keeping up with this increasing wireless data demand, leading to congestion in the network, especially in areas of dense population. As a result, users' data rates have to be throttled to ease congestions [2], [6], [9], at the cost of the degraded quality of service (QoS).

Admittedly, the capacity shortfall of cellular networks can be mitigated by allocating more wireless spectrum and deploying more wireless infrastructures including more and smaller cells and WiFi networks offloading, etc. However, spectrum allocation and infrastructure upgrading are not only costly but also time-consuming, while WiFi networks may not always be available and secure. A promising alternative, inspired by the similar problem of demand response in power networks, is to improve spectrum and infrastructure efficiency through managing wireless data traffic (i.e., demand). Notice that wireless traffic or demand usually fluctuates with a large peak-to-valley ratio throughout a day; see Fig. 1 for a trace of smartphone web browsing activity over a day. However, wireless capacity needs to be provisioned to meet the peak demand rather than the average. This means that the cellular network is usually stressed in peak hours while largely underutilized at other times. If the demand profile can be shaped to reduce the peak and smooth the time variation, not only can more traffic be accommodated under limited existing capacity constraints, but also additional spectrum allocation and infrastructure upgrades can be slowed down, which together greatly improve wireless network efficiency and QoS, and yield huge savings for service providers.

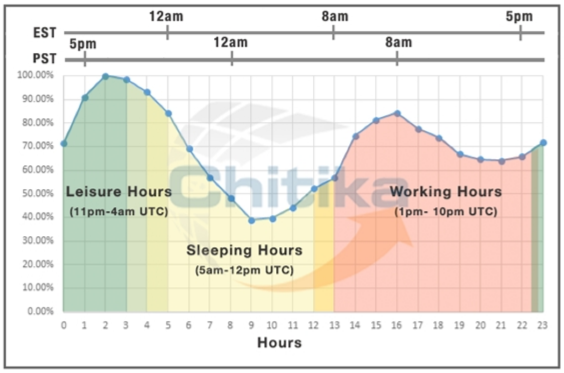


Fig. 1: North America smartphone web browsing activity in one day [21].

In this paper, we focus on designing demand shaping algorithms for cellular networks. We divide wireless traffic into two categories: non-deferrable traffic and deferrable traffic. Non-deferrable traffic refers to the traffic of those applications such as online gaming that have no or low delay tolerance, and constitutes the base traffic whose profile cannot be shaped. Deferrable traffic refers to the traffic of those applications such as file uploading/downloading that are flexible in time and only require being served by a designated deadline, e.g., finishing photo backup on cellphone by 12 am. Deferrable applications (DAs) are further divided into two major types: (1) continuous-rate interruptible applications such as photos backup and applications update that allow any data rates—e.g., the *delayed offloading* in [27], [30], and (2) discrete-rate non-interruptible applications such as online movie streaming and video conference that usually require certain constant data rate [3], [4] and should not be interrupted once started, e.g., one can schedule movie watching or video conference to the "valley" time to enjoy better graphic quality and incur less data cost if he/she has the time flexibility. See Table I for a summary of traffic types and examples. We seek to schedule the deferrable traffic to flatten the aggregate traffic profile over a day.

Specifically, we formulate the cellular traffic demand shaping as an optimization problem that minimizes the (time)

X. Zhou and L. Chen are with College of Engineering and Applied Science, University of Colorado, Boulder, CO 80309, USA (emails: {xinyang.zhou, lijun.chen}@colorado.edu).

Preliminary result of this paper has been presented at the Allerton Conference on Communication, Control, and Computing, Monticello, Illinois, 2014 [39].

Traffic/Application Type	Examples
Non-deferrable application	Online gaming, web browsing
Discrete-rate non-interruptible DA	Movie streaming, video conference
Continuous-rate interruptible DA	Applications update, photos backup

TABLE I: Traffic/Application types and examples.

variation in the aggregate traffic profile subject to the time and rate specification on each DA. We first assume complete traffic information and design an offline demand shaping algorithm. There are two challenging issues in the offline algorithm design. First, the optimization problem is non-convex because of discrete-rate non-interruptible applications. We instead solve its convex relaxation and design a randomized scheme based on the solution to the relaxed problem. Second, demand shaping involves potentially a huge number of applications and users. A centralized algorithm is not scalable. We instead design an iterative and distributed algorithm based on the descent method. We establish the almost surely convergence for the algorithm based on supermartingale theory.

We then consider a more realistic setting with incomplete information where we can only predict future traffic to a certain degree of accuracy, and design an online and distributed demand shaping algorithm that updates the schedules of DAs each timeslot when new information and updated prediction are available, based on the offline algorithm for an optimization problem over a shrinking horizon from the current time to the end of the day. We compare the performance of the online algorithm against the optimal offline algorithm, and provide numerical examples to complement the theoretical analysis.

The rest of the paper is organized as follows. Section II briefly reviews some related work and discusses some related issues. Section III describes the system model and problem formulation. Section IV presents an offline distributed algorithm for demand shaping under the assumption of complete traffic information and characterizes its performance. Section V considers a realistic setting of incomplete traffic information, and presents an online algorithm for demand shaping. Section VI provides numerical examples to complement theoretical analysis, and Section VII concludes the paper.

II. Related Work and Issues

Demand shaping in cellular networks is similar to demand response in power networks, in terms of design objectives, problem formulation, and the associated algorithmic challenges. Indeed, we borrow insights from demand response in power networks; see, e.g., [12], [14], [15], [29]. In particular, our online demand shaping algorithm is motivated by the solution approach for online control of continuous load in reference [15], and mathematically can be seen as its extension to incorporate discrete decision variables considered in reference [14]. However, our model captures realistic cellular traffic settings, as it includes both continuous and discrete decision variables. Moreover, the integration of discrete decision variables into the online algorithm makes the performance analysis of the algorithm more challenging, compared to that in [15]. Related work also includes Zhao *et al* [38] that designs a centralized online EV charging algorithm to minimize the peak procurement from the grid under uncertain prediction of future demand and renewable energy supply, and Parise *et al* [31] that proposes a decentralized charging control for EVs to flatten the aggregate power demand profile. They all consider only continuous decision variables.

To ease the stress from high demand in cellular networks, various demand-shaping-based methodologies as well as traffic offloading strategies have been studied in existing literatures. Tadrous *et al* in [36] propose a paradigm to proactively serve peak-hour requests during the off-peak time based on prediction to smoothen the traffic demand over time without changing customers' activity pattern. However, such strategy is limited to routine behaviors only. In [19] Hajiesmaili *et al* introduce an online procurement auction framework to incentivize mobile devices to participate in device-to-device load balancing to offload traffic from one heavy-loaded base station to adjacent idle ones. Besides, WiFi and femtocell offloading of cellular data is another major approach to easing the congestion of cellular networks; see [10], [13], [22], [26], [27], [30] for related works.

In this paper we have focused on designing demand shaping algorithms based on a general and simplified system model. We do not investigate the important practical issues such as the timescale and granularity at which we schedule and reschedule the DAs. We plan to develop a platform to enable automatic demand shaping in the future, and will investigate various practical issues then. Also, demand shaping involves not only the design of control algorithms but also the design of right mechanisms to incentivize the users to move out of their "comfortable zone" in wireless applications and data usage. Incentive design for demand shaping is currently an active research area; see, e.g., the smart data pricing in wireless networks [18], [35], [37], pricing design in general network service to remove congestions [23], [32], pricing/reward signals in power distribution system [28], [40], and the references therein.

Some discussion on the practicality of demand shaping is also in place. People tend to use mobile data services whenever they want, regardless of whether it is at peak time or valley time for the cellular network. However, a survey [17] conducted in India and USA in 2012 shows that, given proper monetary incentive, many people are willing to postpone their mobile data usage, with acceptable postponement varying from minutes to hours, depending on different types of services and different individual preferences [18]. For example, wireless service providers can motivate the users to shift their demand by implementing the time-dependent pricing (TDP) strategy. TDP is now applied as a simple two-period plan by many wireless service providers around the world, in voice services and data services; e.g., Verizon [8] and Sprint [5] in the US have "happy hours" in the night and weekend for voice service, TelCom [7] in South Africa has "Night Surfer" plans giving free data from 11pm to 5am, and Airtel [1] in India provides unlimited data in the night. More refined TDP strategies can be applied to maximize benefits for both wireless service providers and users, by dynamically adjusting prices according to the data usage of the current time and predicted future. For instance, Ha *et al* [18] have worked on a TDP-based application named TUBE. Trials in cooperation with a local

t	time index, $t \in \mathcal{T} := \{1, \ldots, T\}$
n	DA index, $n \in \mathcal{N} := \{1, \cdots, N\}$
$\mathcal{N}'$	set of N' continuous DAs
$\mathcal{N}''$	set of $N'' = N - N'$ discrete DAs
$\hat{\mathcal{N}}''_t$	set of discrete DAs started earlier
$\tilde{\mathcal{N}}_t$	set of DAs adjustable at time t
b	base traffic profile, $b = \{b(t); t \in \mathcal{T}\}$
p_n	data rate profile of DA n, $p_n = \{p_n(t); t \in \mathcal{T}\}$
$\overline{p}_n(t)$	upper bounds of DA n on the data rate at time t
r_n	constant bit rate for DA $n \in \mathcal{N}''$
l_n	number of timeslots to finish transmission for DA $n \in \mathcal{N}''$
q	virtual deferrable traffic profile
d	average traffic profile
$\hat{d}$	average traffic profile of online ODS
$\hat{d}^*$	average traffic profile of online relaxed ODS
d^*	average traffic profile of offline relaxed ODS
P_n	total traffic required from DA n, $P_n = \sum_{t \in \mathcal{T}} p_n(t)$
$P_n(t)$	remaining traffic to be served for DA $n \in \tilde{\mathcal{N}}_t'$
x_n^k	change in traffic profile of DA n, $x_n^k = p_n^{k+1} - p_n^k$
t_n^a	arrival time of DA n
t_n^d	deadline of DA n
A_n	number of feasible profiles of DA $n \in \mathcal{N}''$
$f_{n,a}$	a-th feasible profile of DA $n \in \mathcal{N}''$
$u_{n,a}$	probability corresponding to $f_{n,a}$
$\mathcal{F}_n$	set of all feasible traffic profiles for discrete DAs, $\mathcal{F}_n = \{f_{n,a}; 1 \le a \le A_n\}$
$V(d)$	objective value: (time) variance of d

TABLE II: Main notation.

wireless service provider shows its effectiveness in shaping the traffic profile [24]. Also refer to [34] for a review of pricing strategies.

III. System Model and Problem Formulation

Consider a cellular network that serves users for different applications such as web browsing, file sharing, real-time entertainment, etc. The applications can be broadly divided into two categories: deferrable applications (DAs) and non-deferrable applications (non-DAs). DAs refer to those applications that are flexible in the starting time and/or data rate, while the non-DAs refer to those that should be served immediately and often have stringent data rate requirement. Please refer to the third paragraph of Section I and TABLE I for more detailed description and examples of DAs and non-DAs.

This work aims to schedule the traffic of DAs so as to flatten the aggregate traffic profile over a day, subject to the time constraints and rate constraints of each application. We use a discrete-time model where one day is divided equally into T timeslots, indexed by $t \in \mathcal{T} = \{1, 2, \cdots, T\}$. The duration of a timeslot can be, e.g., 30 minutes or 1 hour [18], depending on the time resolution of scheduling decisions.

A. Non-Deferrable Applications

Non-DAs include web browsing, online gaming, and real-time chatting with multimedia, etc. The latency tolerated by these applications usually varies from hundreds of milliseconds to seconds. Since these applications should be served immediately upon request, their traffic is inelastic and constitutes the *base traffic* whose profile cannot be shaped. Denote the base traffic profile by $b = \{b(t); t \in \mathcal{T}\}$. As we can only predict the base traffic to a certain accuracy, we model it as a random vector with mean $\overline{b} = \{\overline{b}(t); t \in \mathcal{T}\}$ and random derivation $\delta b = \{\delta b(t); t \in \mathcal{T}\}$ from the mean, i.e., $b = \overline{b} + \delta b$. We assume that $\delta b(t)$ has a mean of 0 and variance of $\delta^2(t)$, and may be temporally correlated. We further assume that we can make better prediction for the timeslots that are closer to current time, modeled by a time-dependent deviation from the mean, i.e., the base traffic at some future time $\tau \in \mathcal{T}$ is predicted at current time t by

$$b_t(\tau) = \overline{b}(\tau) + \delta b_t(\tau), \tag{1}$$

where the subscript t represents the timeslot when the prediction is made, and $\delta b_t(\tau)$ has a decreasing variance $\delta_t^2(\tau)$ as t approaches τ. More concrete model for prediction will be introduced in Section VI. The parameters $\overline{b}$ and δ_t will be specified exogenously, and can be estimated from the historical traffic records.

B. Deferrable Applications

Assume that there are N DAs in the network, indexed by $n \in \mathcal{N} = \{1, \cdots, N\}$. Each DA n is characterized by an arrival time t_n^a when it is requested or after which it can be started, a deadline t_n^d by which its transmission must be done, and certain requirement or constraint on data rate $p_n = \{p_n(t); t \in \mathcal{T}\}$. Let P_n denote the total traffic required by DA n, i.e., $\sum_{t \in \mathcal{T}} p_n(t) = P_n$. We can classify DAs into two main categories: *continuous-rate interruptible* DAs (or continuous DAs for simplicity) that allow any data rates between certain upper and lower bounds and can be interrupted and resumed at any time before the deadline, and *discrete-rate non-interruptible* DAs (or discrete DAs for simplicity) that require certain (roughly) constant data rate and cannot be interrupted once they are started. For example, system backup is usually interruptible and allows any continuous data rates, while video conference is usually preferred to be non-interruptible and runs at a constant (thus discrete) data rate once it is started.

Among the total N DAs, we assume there are N' continuous DAs, indexed by $n \in \mathcal{N}' = \{1, \cdots, N'\}$. For each continuous DA, denote by $\underline{p}_n(t)$ and $\overline{p}_n(t)$ the lower and upper bounds on its data rate at time $t \in \mathcal{T}$, i.e.,

$$\underline{p}_n(t) \le p_n(t) \le \overline{p}_n(t), \; t \in \mathcal{T}. \tag{2}$$

Naturally, $0 \le \underline{p}_n(t) \le \overline{p}_n(t)$. The lower bounds $\underline{p}_n(t)$ are usually zero, and the upper bounds $\overline{p}_n(t)$ can be set according to, e.g., the available bandwidth. The arrival time t_n^a and the deadline t_n^d can be integrated into the rate constraints (2) by setting $\overline{p}_n(t) = 0$ for $t < t_n^a$ and $t > t_n^d$, i.e., no traffic is transmitted before arrival time or after deadline.

Index the rest $N'' = N - N'$ discrete DAs by $n \in \mathcal{N}'' = \{N' + 1, \cdots, N\}$. For a discrete DA such as a streaming application, a constant bit rate r_n corresponds to a certain graphic quality, e.g., $r_n = 3$ Mbps for a SD quality movie on Netflix [4], and $r_n = 1.2$ Mbps for a HD video call on Skype [3]. As the graphic quality usually (preferrably) does not change during those applications, this seemingly over-simplified assumption of a single discrete rate is reasonable.

For each DA $n \in \mathcal{N}''$ with its total traffic P_n and the rate r_n, it takes $l_n = P_n / r_n$ consecutive timeslots (or equivalently the other way around, i.e., we calculate $P_n = l_n * r_n$ based on l_n and r_n). Therefore, the number of its feasible traffic profiles is $A_n = t_n^d - t_n^a - l_n + 1$, wherein the a-th feasible profile is denoted as

$$f_{n,a} = \Big\{ p_n \Big| p_n(t) = \Big\{ \begin{array}{ll} r_n, \text{ if } t_n^a + a - 1 \le t \le t_n^a + a + l_n \\ 0, \qquad\qquad \text{otherwise} \end{array} \Big\}.$$

We denote the set of all feasible traffic profiles of DA $n \in \mathcal{N}''$ by $\mathcal{F}_n = \{f_{n,a} :\ 1 \le a \le A_n\}$, i.e., $p_n \in \mathcal{F}_n$, $\forall n \in \mathcal{N}''$.

Remark 1: All the modeled traffic parameters can be reasonably accessed or estimated in practice. For example, information regarding total required traffic P_n and video streaming rate r_n is available from metadata of traffic to be transmitted, parameters like t_n^a and t_n^d are specified by the users in advance (and $\mathcal{F}_n$ can then be calculated accordingly), whereas data rate bounds $\underline{p}_n(t)$ and $\overline{p}_n(t)$ can be either determined by available bandwidth or designated by the users. See, e.g., [18] for an example system involving similar information requirement and implemented with real users and service provider. □

C. Problem Formulation

We aim to schedule the traffic of DAs, so as to flatten the aggregate traffic profile as much as possible. Denote the "average" traffic profile by $d = \{d(t);\ t \in \mathcal{T}\} := \frac{1}{N}(b + \sum_{n\in\mathcal{N}} p_n)$. Traffic flattening can be achieved by minimizing the time variance of d, formulated as the following optimal demand shaping (**ODS**) problem:

ODS:

$$\min_{p,d} \quad V(d) = \frac{1}{T}\sum_{t\in\mathcal{T}} \big(d(t) - \frac{1}{T}\sum_{\tau\in\mathcal{T}} d(\tau)\big)^2 \qquad (3a)$$

$$\text{s.t.} \quad d(t) = \frac{1}{N}\big(b(t) + \sum_{n\in\mathcal{N}} p_n(t)\big),\ t \in \mathcal{T}, \qquad (3b)$$

$$\underline{p}_n(t) \le p_n(t) \le \overline{p}_n(t),\ t \in \mathcal{T},\ n \in \mathcal{N}', \qquad (3c)$$

$$\sum_{t\in\mathcal{T}} p_n(t) = P_n,\ n \in \mathcal{N}', \qquad (3d)$$

$$p_n \in \mathcal{F}_n,\ n \in \mathcal{N}''. \qquad (3e)$$

Notice that the constraints (3e) for discrete DAs are non-convex. In next section, we will investigate an *offline* algorithm together with a randomized scheme for solving the ODS problem under the assumption of complete information on the base traffic and DAs. Then in Section V, we will study an *online* algorithm for demand shaping under a more realistic setting of incomplete information where we can only predict the future traffic to a certain degree of accuracy. The offline ODS problem and algorithm will later serve as a benchmark to characterize the performance of the online algorithm.

IV. Offline Demand Shaping Algorithm

In this section, we assume complete traffic information, i.e., the base traffic and arrival of DAs are accurately known, and study how to solve the resulting **offline ODS** problem. The offline problem and algorithm will provide insights into the online algorithm design for a realistic setting of incomplete information that will be considered in Section V.

A. Convex Relaxation and Randomized Scheme

The offline ODS problem is non-convex, as each discrete DA has to pick a traffic profile from a discrete set; see constraint (3e). Consider the convex hull of $\mathcal{F}_n$, defined as

$$\begin{aligned} \text{conv}(\mathcal{F}_n) \quad := \quad & \Big\{p_n |\ p_n = \sum_{a=1}^{A_n} u_{n,a}\cdot f_{n,a},\ u_{a,n} \ge 0 \\ & \text{and } \sum_{a=1}^{A_n} u_{n,a} = 1\Big\}, \end{aligned} \qquad (4)$$

where $u_n := \{u_{n,1}, \ldots, u_{n,A_n}\}$ is the convex combination coefficients, and will be interpreted as probability distribution in the randomized algorithm to be introduced soon. We will instead solve the convex relaxation of the ODS problem by replacing (3e) with the following constraint:

$$p_n \in \text{conv}(\mathcal{F}_n),\ n \in \mathcal{N}''. \qquad (5)$$

We call the relaxed problem (3a)–(3d)(5) the **R-ODS** problem. However, a solution $p_n^* \in \text{conv}(\mathcal{F}_n),\ n \in \mathcal{N}''$ to the R-ODS problem might not be feasible for original ODS, i.e., $p_n^* \notin \mathcal{F}_n$. But since by definition (4) a solution p_n^* can always be written as the convex combination $\sum_{a=1}^{A_n} u_{n,a} f_{n,a}$ we will randomly pick a traffic profile $p_n = f_{n,a} \in \mathcal{F}_n$ with corresponding probability $u_{n,a}$. That said, we will design a randomized algorithm for the offline ODS problem, based on the solution to the R-ODS problem. We will integrate it into a distributed algorithm next.

B. Distributed Algorithm

Solving the R-ODS problem (and the offline ODS problem) directly in a centralized way requires collecting information on all DAs, which may incur too much communication overhead and is impractical in the real network. Moreover, the users may not be willing to reveal information on DAs due to privacy concern. Therefore, we seek to solve it in a distributed way. Noticing that R-ODS problem has decoupled constraints, we attempt to design an *iterative and distributed* algorithm based on the decent method [11].

Before deriving the algorithm, we establish the following useful results. At k-th iteration, let $p^k = \{p_n^k;\ n \in \mathcal{N}\}$ be the traffic profiles of all DAs, $d^k = \frac{1}{N}(b + \sum_{n\in\mathcal{N}} p_n^k)$ the average traffic profile, and $x_n^k = p_n^{k+1} - p_n^k,\ n \in \mathcal{N}$ the change in traffic profile of DA n from iteration k to $k+1$. We have:

$$E\big[\| \sum_{n\in\mathcal{N}} x_n^k \|_2^2\big] = \sum_{n\in\mathcal{N}} Var(x_n^k) + \| \sum_{n\in\mathcal{N}} E[x_n^k] \|_2^2, \qquad (6)$$

where the variance $Var(x_n^k) := E\big[\|x_n^k\|_2^2\big] - \|E[x_n^k]\|_2^2$, and $E[\cdot]$ denotes the average.[1] By Jensen's inequality,

$$\| \sum_{n\in\mathcal{N}} E[x_n^k] \|_2^2 \le N \sum_{n\in\mathcal{N}} \big\|E[x_n^k]\big\|_2^2. \qquad (7)$$

Therefore, one has

$$E\big[\| \sum_{n\in\mathcal{N}} x_n^k \|_2^2\big] \le \sum_{n\in\mathcal{N}} Var(x_n^k) + N \sum_{n\in\mathcal{N}} \big\|E[x_n^k]\big\|_2^2. \qquad (8)$$

[1] Notice that we consider a randomized scheme only for discrete DAs. That said, for continuousDAs there is no randomness and their variance is zero.

And it follows that

$$\begin{aligned} & TN^2\big(E[V(d^{k+1})|p^k] - V(d^k)\big) \\ = \; & E\big[\| \sum_{n\in\mathcal{N}} x_n^k \|_2^2 + 2\langle Nd^k, \sum_{n\in\mathcal{N}} x_n^k\rangle\big] \\ \leq \; & \sum_{n\in\mathcal{N}} Var(x_n^k) + N\sum_{n\in\mathcal{N}} \|E[x_n^k]\|_2^2 + 2\sum_{n\in\mathcal{N}} E[\langle Nd^k, x_n^k\rangle] \\ = \; & \sum_{n\in\mathcal{N}'} \big(2\langle Nd^k, x_n^k\rangle + N\|x_n^k\|_2^2\big) + \sum_{n\in\mathcal{N}''} \big(2\langle Nd^k, E[x_n^k]\rangle \\ & + N\|E[x_n^k]\|_2^2 + Var(x_n^k)\big). \qquad (9) \end{aligned}$$

Denote by W_1 the first term in (9) and W_2 the second. For $n \in \mathcal{N}'$, we choose p_n^{k+1} so as to minimize W_1, i.e., to solve

$$\begin{aligned} \min_{p_n} \quad & 2\langle d^k, p_n - p_n^k\rangle + \|p_n - p_n^k\|_2^2 & (10a) \\ \text{s.t.} \quad & (3c) - (3d). & (10b) \end{aligned}$$

On the other hand, after some mathematical manipulations, we have

$$W_2 = \sum_{n\in\mathcal{N}''} \big(2N\langle d^k - p_n^k, E[p_n^{k+1}]\rangle + (N-1)\|E[p_n^{k+1}]\|_2^2\big) + \Pi^k,$$

where Π^k is a constant given p_n^k. For $n \in \mathcal{N}''$, we choose p_n^{*k+1} so as to minimize W_2, i.e., to solve

$$\min_{p_n\in \text{conv}(\mathcal{F}_n)} 2\langle d^k - p_n^k, p_n\rangle + \frac{N-1}{N}\|p_n\|_2^2. \qquad (11)$$

In essence, what we have done is to *maximize the expected incremental decrease in the objective value* $V(d)$ at each iteration (i.e., steepest descent). This motivates a distributed demand shaping algorithm with the collaboration of a coordinator; see Algorithm 1. The wireless service provider can implement a logical coordinator at the base station.

Algorithm 1 Offline Demand Shaping (Off-DS) Algorithm

```
At $k$-th iteration:
1) Upon gathering traffic profiles $p_n^k$ from DAs, the coordinator calculates the average traffic profile $d^k = \frac{1}{N}(b + \sum_{n\in\mathcal{N}} p_n^k)$ and announces it to DAs (or the end users) over a signaling or control channel.
2) Upon receiving the average traffic profile $d^k$,
   • DA $n \in \mathcal{N}'$ updates its traffic profile by
         $p_n^{k+1} = \arg\min_{p_n} \; \|p_n - p_n^k + d^k\|_2^2$
                      s.t. (3c)–(3d),
     and submits it to the coordinator.
   • DA $n \in \mathcal{N}''$ calculates the average traffic profile by
         $p_n^{*k+1} = \arg\min_{p_n\in\text{conv}(\mathcal{F}_n)} \Big\|p_n - \frac{N}{N-1}(p_n^k - d^k)\Big\|_2^2,$
     which is $p_n^{*k+1} = \sum_{a=1}^{A_n} u_{n,a}^{k+1} f_{n,a}$, and then randomly chooses a traffic profile $p_n^{k+1} = f_{n,a}$ with probability $u_{n,a}^{k+1}$ and submits it to the coordinator.
```

The Off-DS algorithm is a distributed algorithm wherein each DA solves its own simple optimization problem based on its previous decision, the average traffic profile d^k, and local constraints, while the coordinator collects the proposed traffic profiles and updates the average traffic profile. Therefore, this algorithm is not only preserving privacy of the users, but also scalable and thus capable of quick response, which is crucial especially in real-time implementation in Section V.

The computational complexity of the Off-DS algorithm is estimated as follows for completeness. Given certain accuracy requirement $\epsilon > 0$ in the objective function value, the descent method requires $O(\log(1/\epsilon))$ iterations [11]. At each iteration, DAs solves an easy quadratic programming with a polynomial complexity of $O(T^{O(1)})$ [33]. On the other hand, the coordinator calculates the average traffic profile which requires $O(N)$ complexity each iteration. As a result, the Off-DS algorithm requires overall computational complexity of $O\big((N + T^{O(1)})\log(1/\epsilon)\big)$.

Remark 2: For simpler expression, we use p_n as the decision variable for DA $n \in \mathcal{N}''$ in algorithm design and analysis, while in real implementation, it is more convenient to use probability distribution u_n as the equivalent decision variable. Also notice that, if there is no continuous DA, Algorithm 1 reduces to the stochastic algorithm in [14]. We expect that the solution approach—randomized algorithm based on the "steepest" descent method for the convex relaxed problem—that we lay out in Sections IV-A and IV-B will find broad application in designing efficient algorithms for optimization problems that involve both continuous and discrete decision variables. □

C. Convergence

Before showing the convergence of the Off-DS algorithm, we first establish two useful relations. For each DA $n \in \mathcal{N}'$, since p_n^{k+1} solves the problem (10), we have the first-order optimality condition

$$\langle p_n^{k+1} - p_n^k + d^k, p_n - p_n^{k+1}\rangle \geq 0 \qquad (12)$$

for any feasible p_n. Set $p_n = p_n^k$ to obtain

$$\langle d^k, p_n^{k+1} - p_n^k\rangle \leq -\|p_n^{k+1} - p_n^k\|_2^2. \qquad (13)$$

For each DA $n \in \mathcal{N}''$, recalling that $p_n^{*k+1} = E[p_n^{k+1}]$, by the first-oder optimality condition, we have

$$\langle \frac{N}{N-1}(d^k - p_n^k) + p_n^{*k+1}, p_n - p_n^{*k+1}\rangle \geq 0 \qquad (14)$$

for any feasible p_n. Set $p_n = p_n^k$ to get

$$\begin{aligned} \langle Nd^k, p_n^{*k+1} - p_n^k\rangle \;\leq\; & -(N-1)\|p_n^{*k+1} - p_n^k\|_2^2 \\ & + \langle p_n^k, p_n^{*k+1} - p_n^k\rangle. \qquad (15) \end{aligned}$$

Now, construct a filtration Σ^* of the probability space $\{\Omega, \Sigma, \mathcal{P}\}$, where the sample space Ω is the feasible set specified by the constraints (3c)–(3e), the σ-algebra $\Sigma_k = \Omega,\ k \geq 0$, and $\mathcal{P}(\Sigma_k) = \{\delta(p_n - p_n^k), n \in \mathcal{N}';\ u_{n,a}^k, 1 \leq a \leq A_n, n \in \mathcal{N}''\}$, i.e., determined by the k-th iteration of the Off-DS algorithm.

Theorem 1: The pair $(V(d),\ \Sigma^*)$ is a supermartingale. □

Proof: First, notice that $V(d)$ is bounded from below. So, $E[-\min\{0, V(d)\}] < \infty$. Second, applying relations (13)–

(15) to equation (9), we obtain

$$\begin{aligned} & TN^2\big(E[V(d^{k+1})|p^k] - V(d^k)\big) \\ \leq\ & \sum_{n\in\mathcal{N}'} -N\|x_n^k\|_2^2 + \sum_{n\in\mathcal{N}''} \big(Var(x_n^k) \\ & \quad + (-N+2)\big\|E[x_n^k]\big\|_2^2 + 2\langle p_n^k, p_n^{*k+1} - p_n^k\rangle\big) \\ =\ & \sum_{n\in\mathcal{N}'} -N\|x_n^k\|_2^2 + \sum_{n\in\mathcal{N}''} (-N+1)\big\|E[x_n^k]\big\|_2^2 \\ \leq\ & 0, \end{aligned}$$

i.e., $E[V(d^{k+1})|p^k] \leq V(d^k)$. By definition, $(V(d),\ \Sigma^*)$ is a supermartingale [16]. ■

Notice that $(V(d),\ \Sigma^*)$ is a nonnegative supermartingale. By the martingale convergence theorem [16], the following result is immediate.

Corollary 1: $V(d^\infty) = \lim_{k\to\infty} V(d^k)$ exists almost surely, where $V(d^\infty)$ is some random variable. □

Theorem 2: Denote by $\mathcal{P}^\infty$ an "equilibrium" distribution over traffic profiles that $(V(d),\ \Sigma^*)$ converges to. The support of $\mathcal{P}^\infty$ is a singleton. □

Proof: When $(V(d),\ \Sigma^*)$ converges, $E[V(d^{k+1})|p^k] = V(d^k)$. This requires $E[x_n^k] = E[x_{n'}^k],\ n, n' \in \mathcal{N},\ p_n^{k+1} = p_n^k,\ n \in \mathcal{N}'$, and $p_n^{*k+1} = p_n^k,\ n \in \mathcal{N}''$ for (8), (13), and (15) to hold with equality. Notice that $p_n^{*k+1} = p_n^k$ implies $p_n^{k+1} = p_n^k$, as different feasible traffic profiles of DA $n \in \mathcal{N}''$ are linearly independent. Thus, $p_n^{k+1} = p_n^k,\ n \in \mathcal{N}$. So, the support of $\mathcal{P}^\infty$ contains only one point. ■

Denote by p^∞ an "equilibrium" traffic profile of the Off-DS algorithm, i.e., if $p^k = p^\infty$, then $p^{k+1} = p^\infty$. Obviously the set of equilibrium profiles is not empty, as an optimum of the offline ODS problem is an equilibrium. The following result follows immediately from Theorem 2 and Corollary 1.

Theorem 3: The Off-DS algorithm converges almost surely to an equilibrium traffic profile. □

By equations (12)–(14), we have the following optimality conditions at equilibrium p^∞: for any feasible p_n,

$$\langle b + \sum_{m\in\mathcal{N}} p_m^\infty, p_n - p_n^\infty\rangle \geq 0,\ n \in \mathcal{N}', \tag{16a}$$

$$\langle b + \sum_{m\neq n} p_m^\infty, p_n - p_n^\infty\rangle \geq 0,\ n \in \mathcal{N}''. \tag{16b}$$

D. Performance Analysis of the Offline Algorithm

We now characterize the performance of Off-DS algorithm with respect to the relaxed problem R-ODS that at optimum may attain a lower objective value than the ODS problem. Specifically, denote by p^* the solution of R-ODS, we bound the gap between the equilibrium of the Off-DS algorithm and the solution of the R-ODS problem as: $G^{\text{off}} := V(d^\infty) - V(d^*)$, where $d^\infty = (b + \sum_{n\in\mathcal{N}} p_n^\infty)/N$ and $d^* = (b + \sum_{n\in\mathcal{N}} p_n^*)/N$. Denote by $G_r^{\text{off}} := (V(d^\infty) - V(d^*))/V(d^*)$ the relative gap achieved by the Off-DS algorithm.

Theorem 4: For the Off-DS algorithm, the gap G^{off} is bounded as follows:

$$G^{\text{off}} \leq \frac{2}{TN^2} \sum_{n\in\mathcal{N}''} \|p_n^\infty\|_2^2. \tag{17}$$

Moreover, the relative gap diminishes as the number N'' of discrete DAs increases, i.e.,

$$\lim_{N''\to\infty} G_r^{\text{off}} = 0. \tag{18}$$

□

Proof: For notational simplicity, let $c_d := \sum_{t\in\mathcal{T}} d(t)/T$, which is a constant given the total amount of traffic. The objective value can be written as

$$\begin{aligned} V(d) &= \frac{1}{T}\|d - c_d \cdot \mathbf{1}\|_2^2 = \frac{1}{T}(\|d\|_2^2 + c_d^2\|\mathbf{1}\|_2^2 - 2\langle d, \mathbf{1}\rangle) \\ &= \frac{1}{T}(\|d\|_2^2 + T\cdot c_d^2 - 2T\cdot c_d), \end{aligned}$$

where only the part $\|d\|_2^2$ contains decision variables. We can thus write the gap G^{off} as

$$\begin{aligned} & G^{\text{off}} = V(d^\infty) - V(d^*) = \frac{1}{T}\big(\|d^\infty\|_2^2 - \|d^*\|_2^2\big) \\ =\ & \frac{1}{T}\big(- \|d^\infty - d^*\|_2^2 + \langle 2d^\infty, d^\infty - d^*\rangle\big) \\ \leq\ & \frac{1}{T}\langle 2d^\infty, d^\infty - d^*\rangle \\ =\ & \frac{2}{N^2}\Big(\sum_{n\in\mathcal{N}'}\langle Nd^\infty, p_n^\infty - p_n^*\rangle + \sum_{n\in\mathcal{N}''}\langle Nd^\infty, p_n^\infty - p_n^*\rangle\Big) \\ \leq\ & \frac{2}{TN^2}\sum_{n\in\mathcal{N}''}\langle p_n^\infty, p_n^\infty - p_n^*\rangle \\ \leq\ & \frac{2}{TN^2}\sum_{n\in\mathcal{N}''}\|p_n^\infty\|_2^2, \end{aligned}$$

where the second inequality follows from (16). Note that $\|p_n^\infty\|_2^2$ is a constant for $n \in \mathcal{N}''$. Then the relative gap G_r^{off} can be bounded as

$$\begin{aligned} G_r^{\text{off}} &\leq \frac{2}{TN^2}\sum_{n\in\mathcal{N}''}\|p_n^\infty\|_2^2/V(d^*) \\ &= \frac{\sum_{n\in\mathcal{N}''}\|p_n^\infty\|_2^2}{\|b + \sum_{n\in\mathcal{N}} p_n^*\|_2^2 + N^2(T\cdot c_d^2 - 2T\cdot c_d)}, \end{aligned} \tag{19}$$

whose numerator increases linearly with N'' and denominator increases linearly with the square of N''. Equation (18) follows. ■

Remark 3: We use the relaxed problem R-ODS for comparison instead of the ODS problem for two reasons. First, it is difficult to characterize the optimum of the non-convex ODS problem, and thus evaluating the gap between the equilibrium of the Off-DS algorithm and the optimum of ODS problem is mathematically hard. Second, R-ODS achieves an optimal objective value that is not greater than ODS, resulted from convex relaxation for the discrete decision variables. Therefore, G^{off} provides an upper bound for the "actual" sub-optimality, i.e., the gap between the equilibrium of Off-DS and the optimum of ODS. □

V. Online Demand Shaping Algorithm

In this section, we consider a realistic setting with incomplete information where we can only predict future traffic to a certain degree of accuracy, and study online demand shaping that makes decisions based on the prediction of future traffic and updates the decision as new information becomes available.

A typical algorithm used in this setting is the receding horizon control; see, e.g., [25]. However, as the objective function (3a) does not have a nice additive structure, receding horizon control algorithm does not admit an easy analysis. We will instead extend a shrinking horizon control algorithm, which is used in [15] that studies mathematically the same problem with only continuous DAs, to include discrete DAs, and apply it to our online demand shaping (**online DS**) problem.

A. *Online Algorithm*

We assume that the number m_t of DAs arriving at time t is randomly distributed with a mean λ_t and variance $(\delta\lambda_t)^2$, and the total amount of traffic of each DA is randomly distributed with a mean P and variance $(\delta P)^2$. Denote by $\mathcal{N}'_t = \{1, \cdots, N'_t\}$ the set of continuous DAs and $\mathcal{N}''_t = \{N'+1, \cdots, N_t\}$ the set of discrete DAs that have arrived by time $t \in \mathcal{T}$, and let $\mathcal{N}_t = \mathcal{N}'_t \cup \mathcal{N}''_t$ and $N''_t = N_t - N'_t$. Notice that we cannot reschedule the remaining traffic of a discrete DA that has already started. Denote by $\tilde{\mathcal{N}}''_t \subseteq \mathcal{N}''_t$ the set of discrete DAs that have not been started by time t. For DA $n \in \tilde{\mathcal{N}}''_t$, denote by $\mathcal{F}_n(t) = \{f_{n,a}; 1 \le a \le A_n(t)\}$ the set of feasible traffic profiles at time t. Let $\tilde{\mathcal{N}}_t = \mathcal{N}'_t \cup \tilde{\mathcal{N}}''_t$ be the set of DAs whose profiles are still adjustable at time t (i.e., all the continuous DAs and the discrete DAs that have not started by time t).

At time t, we make a prediction $b_t(t:T)$ of base traffic for the rest timeslots of the day, and we also have the information on DA $n \in \mathcal{N}_t$ and the expected total future deferrable traffic $\sum_{\tau=t+1}^{T} P\lambda_\tau$. Following [15], we introduce a *virtual* deferrable traffic profile $q(t:T) = \{q(\tau); t \le \tau \le T\}$ with $q(t) = 0$ and $\sum_{\tau=t}^{T} q(\tau) = \sum_{\tau=t+1}^{T} P\lambda_\tau$, to emulate the impact of the future deferrable traffic upon the current demand shaping decision. With the afore setup, we aim to schedule and reschedule the DAs, so as to solve the following problem at each timeslot $t \in \mathcal{T}$.

ODS$_t$:

$$\min \; V(d) = \frac{1}{T-t+1}\sum_{\tau=t}^{T}\Big(d(\tau) - \frac{\sum_{s=t}^{T} d(s)}{T-t+1}\Big)^2 \quad (20a)$$

$$\text{over} \;\; p(t:T),\; d(t:T),\; q(t:T)$$

$$\text{s.t.} \;\; d(\tau) = \frac{b_t(\tau)+q(\tau)+\sum_{n\in\mathcal{N}_t} p_n(\tau)}{N_t}, \;\; \tau \ge t, \quad (20b)$$

$$\underline{p}_n(\tau) \le p_n(\tau) \le \overline{p}_n(\tau), \tau \ge t, \;\; n \in \mathcal{N}'_t, \quad (20c)$$

$$\sum_{\tau=t}^{T} p_n(\tau) = P_n(t), \;\; n \in \mathcal{N}'_t, \quad (20d)$$

$$p_n \in \mathcal{F}_n(t), \;\; n \in \tilde{\mathcal{N}}''_t, \quad (20e)$$

$$\sum_{\tau=t}^{T} q(\tau) = \sum_{\tau=t+1}^{T} P\lambda_\tau, \quad (20f)$$

where $p(t:T) = \{p_n(\tau); t \le \tau \le T, n \in \tilde{\mathcal{N}}_t\}$, $d(t:T) = \{d(\tau); t \le \tau \le T\}$, and $P_n(t) = P_n - \sum_{\tau=1}^{t-1} p_n(\tau), n \in \mathcal{N}'_t$ is the amount of traffic to be served at or after time t.

We can solve the ODS$_t$ problem at each timeslot the same way as we solve the offline ODS problem (3), constituting an online demand shaping algorithm; see Algorithm 2, wherein the convergence (and computational complexity) of Step 2) can be established (and analyzed) in the same way as Algorithm 1.

Algorithm 2 Online Demand Shaping (On-DS) Algorithm

```
At each timeslot $t \in \mathcal{T}$:
  1) Denote by $p_n^{(t-1)}, n \in \mathcal{N}_{t-1}$ the schedules determined by time $t-1$, and by $\hat{\mathcal{N}}''_t \subseteq \mathcal{N}''_t$ the set of discrete DAs that has been started before time $t$. For each DA $n \in \hat{\mathcal{N}}''_t$, set its schedule $p_n(t;T) = \{p_n(\tau); t \le \tau \le T\}$ as $p_n(\tau) = p_n^{(t-1)}(\tau), \; t \le \tau \le T$.
  2) Solve the ODS$_t$ problem iteratively: at $k$-th iteration,
     a) Upon gathering traffic profiles $p_n^k(t:T) = \{p_n^k(\tau); t \le \tau \le T\}$ from DAs $n \in \tilde{\mathcal{N}}_t$, the coordinator solves
            $\min_{q(t+1:T)} \sum_{\tau=t+1}^{T} \Big(b_t(\tau) + q(\tau) + \sum_{n\in\hat{\mathcal{N}}''_t} p_n(\tau) + \sum_{n\in\tilde{\mathcal{N}}_t} p_n^k(\tau)\Big)^2$
            s.t.   (20f),
        to obtain a virtual deferrable traffic $\{q^k(\tau); t+1 \le \tau \le T\}$, and then calculates the average traffic $d^k(\tau) = \frac{1}{N_t}\big(b_t(\tau)+q^k(\tau)+\sum_{n\in\hat{\mathcal{N}}''_t} p_n(\tau)+\sum_{n\in\tilde{\mathcal{N}}_t} p_n^k(\tau)\big)$ for $\tau \ge t$ and announces it to DA $n \in \tilde{\mathcal{N}}_t$ over a signaling or control channel.
     b) Upon receiving the average traffic profile $d^k$,
        • DA $n \in \mathcal{N}'_t$ obtains $p_n^{k+1}(t:T)$ by
              $\min_{p_n(t:T)} \; \big\|p_n(t:T) - p_n^k(t:T) + d^k(t:T)\big\|_2^2$
              s.t.  (20c)–(20d),
          and submits the updated profile to the coordinator.
        • DA $n \in \tilde{\mathcal{N}}''_t$ calculates $p_n^{*k+1}(t:T)$ by
              $\min_{p_n(t:T)} \; \Big\|p_n(t:T) - \frac{N_t}{N_t-1}(p_n^k(t:T) - d^k(t:T))\Big\|_2^2$
              s.t.   $p_n(t:T) \in \text{conv}(\mathcal{F}_n(t)), \; n \in \tilde{\mathcal{N}}''_t$,
          represents it as a convex combination $p_n^{*k+1} = \sum_{a=1}^{A_n(t)} u_{n,a}^{k+1} f_{n,a}$, and randomly chooses a traffic profile $p_n^{k+1} = f_{n,a}$ with probability $u_{n,a}^{k+1}$ and submits it to the coordinator.
```

B. *Performance Analysis of the Online Algorithm*

We now characterize the performance of On-DS algorithm with respect to the result of Off-DS algorithm which serves as a benchmark. We will make the following assumptions to simplify the analysis and obtain insights into how uncertainties affect the performance of On-DS algorithm.

Assumption 1: The amount of deferrable traffic is large and flexible enough so that a valley-filling schedule exists at every time $t = 1, \ldots, T$, i.e., there exists some constant $C(t) \ge b_t(\tau), \forall \tau = t, \ldots, T$ such that

$$\begin{aligned} Nd(t) &= C(t) \\ &= \frac{1}{T-t+1}\Big(\sum_{\tau=t}^{T} b_t(\tau) + \sum_{\tau=t+1}^{T} P\lambda_\tau + \sum_{n=1}^{N_t} P_n(t)\Big). \end{aligned} \quad (21)$$

□

Remark 4: Assumption 1 looks a strong assumption, and we do not have empirical evidence to support it as demand shaping has not being widely adopted in current cellular networks. However, with increasing penetration of deferrable traffics and users, this assumption expects to hold. One purpose of algorithm design as in this paper and incentive design

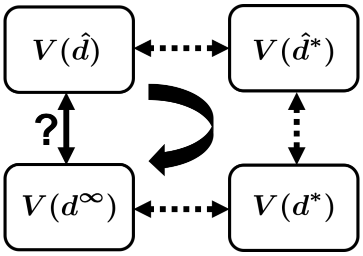


Fig. 2: Strategy to calculate the gap between the equilibrium of the On-DS algorithm and that of Off-DS algorithm.

as in [18] is to facilitate and incentivize wide adoption of demand shaping. On the other hand, valley-filling represents the scenario where demand shaping is most useful and presents a benchmark for the potential of demand shaping. Mathematically, it is very difficult to analyze the performance of the online algorithm under more general assumption than Assumption 1. However, notice that in numerical examples in Section VI, we do not impose Assumption 1 while the results still fall into the bound specified in Theorem 5. □

Assumption 2: The base traffic prediction at t is modeled as the following causal filter

$$b_t(\tau) = \bar{b}(\tau) + \sum_{s=1}^{T} e(s) f(\tau - s),\ \tau = 1, \dots, T, \tag{22}$$

where $e = \{e(s)\}_{s=1}^{T}$ is an uncorrelated sequence of independent and identically distributed random variables with mean 0 and variance δ^2, and $f = \{f(\tau)\}_{\tau=-\infty}^{\infty}$ is the impulse response with $f(0) = 1$. Let $F(t) := \sum_{s=0}^{t} f(s)$. □

We denote by G^{on} the gap defined as the expected difference between the results of On-DS algorithm and Off-DS algorithm, i.e., $G^{\text{on}} = E[V(\hat{d}) - V(d^\infty)]$, where E denotes the expectation, and $\hat{d}$ and d^∞ denote the average traffic profiles achieved by the On-DS algorithm and the offline-DS algorithm respectively. It turns out that direct calculation of this gap is difficult. We therefore utilize two intermediate variables: d^*, the average traffic profile achieved by the R-ODS problem, and $\hat{d}^*$, the average traffic profile achieved by the relaxed online DS, i.e., the counterpart of R-ODS problem in the online scenario. Similar notations are applied to individual traffic profile p_n. With the relation shown in Fig. 2, we can write online gap as

$$\begin{aligned} G^{\text{on}} &= E\big[V(\hat{d}) - V(\hat{d}^*) + V(\hat{d}^*) - V(d^*) + V(d^*) - V(d^\infty)\big] \\ &= E\big[V(\hat{d}) - V(\hat{d}^*)\big] + E\big[V(\hat{d}^*) - V(d^*)\big] \\ &\quad + E\big[V(d^*) - V(d^\infty)\big]. \end{aligned} \tag{23}$$

Theorem 5: The gap, i.e., the expected difference between the results of On-DS algorithm and Off-DS algorithm is bounded as follows:

$$\begin{aligned} G^{\text{on}} = E[V(\hat{d}) - V(d^\infty)] &\le \frac{2}{TN^2} \sum_{n \in \mathcal{N}''} \|\hat{p}_n\|_2^2 \\ &+ \frac{(\delta\lambda)^2}{T} \sum_{t=2}^{T} \frac{1}{t} + \frac{\delta^2}{T^2} \sum_{t=0}^{T-1} F^2(t) \frac{T-t-1}{t+1}. \end{aligned} \tag{24}$$

□

Proof: Applying the approach and results from Theorem 4, we have

$$0 \le E\big[V(\hat{d}) - V(\hat{d}^*)\big] \le \frac{2}{TN^2} \sum_{n \in \mathcal{N}''} \|\hat{p}_n\|_2^2, \tag{25}$$

$$-\frac{2}{TN^2} \sum_{n \in \mathcal{N}''} \|\hat{p}_n\|_2^2 \le E\big[V(d^*) - V(d^\infty)\big] \le 0. \tag{26}$$

For the second term of (23), under Assumptions 1–2, following [15], we get

$$E\big[V(\hat{d}^*) - V(d^*)\big] = \frac{(\delta\lambda)^2}{T} \sum_{t=2}^{T} \frac{1}{t} + \frac{\delta^2}{T^2} \sum_{t=0}^{T-1} F^2(t) \frac{T-t-1}{t+1}. \tag{27}$$

Combine (25)–(27) to obtain (24). ■

Theorem 5 indicates that, the size of the gap between online and offline algorithms changes monotonically with prediction error of both base traffic and future arrival of deferrable traffic. Accordingly we can improve the result of On-DS algorithm by implementing better prediction mechanism, e.g., On-DS algorithm which updates its prediction to keep the value of prediction error small. Also, if the impulse response f is chosen to fade quickly enough, then as we have finer time granularity, we have $T \to \infty$, and $G^{\text{on}} \to 0$, which intuitively indicates that, with infinitely small timeslot, we can update our decisions frequently enough to mitigate prediction errors, and therefore have a negligible performance gap.

Lastly, similar to Theorem 4, define a relative gap $G_r^{\text{on}} := G^{\text{on}} / V(d^\infty)$. The following result is immediate.

Theorem 6: The relative gap G_r^{on} diminishes as the number of discrete DAs N'' increases, i.e.,

$$\lim_{N'' \to \infty} G_r^{\text{on}} = 0. \tag{28}$$

□

Remark 5: It is worth noting that equation (28) does not necessarily imply a monotone decreasing of G_r^{on} with respect to N''. This can be seen from Fig. 6 in Section VI that does not show a decreasing G_r^{on} as N'' increases. □

By equations (25) and (27), it is straightforward to obtain the following result.

Corollary 2: The expected difference between the On-DS algorithm and the optimum of the R-ODS problem is bounded as follows:

$$\begin{aligned} E[V(\hat{d}) - V(d^*)] &\le \frac{2}{TN^2} \sum_{n \in \mathcal{N}''} \|\hat{p}_n\|_2^2 \\ &+ \frac{(\delta\lambda)^2}{T} \sum_{t=2}^{T} \frac{1}{t} + \frac{\delta^2}{T^2} \sum_{t=0}^{T-1} F^2(t) \frac{T-t-1}{t+1}. \end{aligned}$$

□

VI. Numerical Examples

In this section, we provide numerical examples to evaluate the performance of the On-DS algorithm. We use certain composite traffic traces to drive simulations to show the impact of base traffic prediction errors, deferrable traffic prediction errors, and deferrable traffic penetration levels. We expect the conclusions obtained to hold for real traffic.

A. Experimental Setup

Consider a 48-hour period of time starting from 4:00 pm to 3:59 pm two days later. We divide the 48 hours equally into

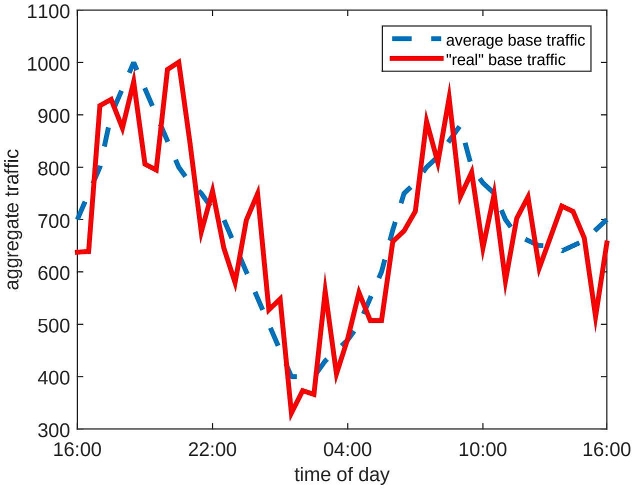


Fig. 3: Base traffic: the average (blue/dotted) and a "real" trace (red/solid).

96 timeslots, each 30 minutes long. We consider scheduling traffic that arrives within the first 24 hours only, which may be allocated to the second 24 hours.

1) Non-deferrable Traffic: The "real" trace we use for non-deferrable traffic, or base traffic, is shown in Fig. 3 (red line). It is constructed by random fluctuation around the average base traffic trace (blue line) composed based on North American mobile web browsing activity by time of day in 2013 [21], shown in Fig. 1. As modeled in Section III-A, the prediction of base traffic follows (1), consisting of average base traffic $\bar{b}(\tau)$ and random deviation $\delta b_t(\tau)$ from the average value. Following [15], at time t, $\delta b_t(\tau)$ is modeled as

$$\delta b_t(\tau) = \sum_{s=t+1}^{\tau} \omega_s(\tau), \ t < \tau \leq T, \tag{29}$$

where $\omega_s(\tau)$ are random variables of Gaussian distribution with 0 mean and variances

$$E[\omega_s^2(\tau)] = \frac{\sigma^2}{\tau - s + 1}, \ 1 \leq s \leq \tau \leq T. \tag{30}$$

In this way, $\delta b_t(\tau)$ has decreasing variance as t approaches τ, simulating a gradually improving prediction for some future timeslot τ as one gets closer to it. In simulation, we take the values of σ^2 in (30) from 0 to 100 with increment of 10, corresponding to a root-mean-square prediction error (RMSE) ranging from 0% to 32%, looking 48 timeslots (24 hours) ahead.

2) Deferrable Traffic: We assume that the number of DAs arriving at each timeslot follows a "shifted" Poisson process $m + \text{poissrnd}(\lambda_p)$, with $m \geq 0$ and $\text{poissrnd}(\lambda_p)$ denoting a Poisson process with rate λ_p. Here, we set $\lambda_p = 4$, while each DA has a 50-50 chance to require continuous- or discrete-rate traffic. The total traffic P_n of each DA is uniformly distributed in $[\underline{P}, \overline{P}]$ where we set $\underline{P} = 12$ and $\overline{P} = 24$. The deadline for DA n is uniformly distributed in $[t_n^a + l_n + \underline{D}, t_n^a + l_n + \overline{D}]$, where $l_n = \ulcorner P_n / \overline{p}_n \urcorner$ is the minimum number of timeslots required by the DA calculated by ceiling function $\ulcorner \cdot \urcorner$. We set $\underline{D} = 6$, $\overline{D} = 14$, and a universal bit rate upper bound $\overline{p}_n = 3$.

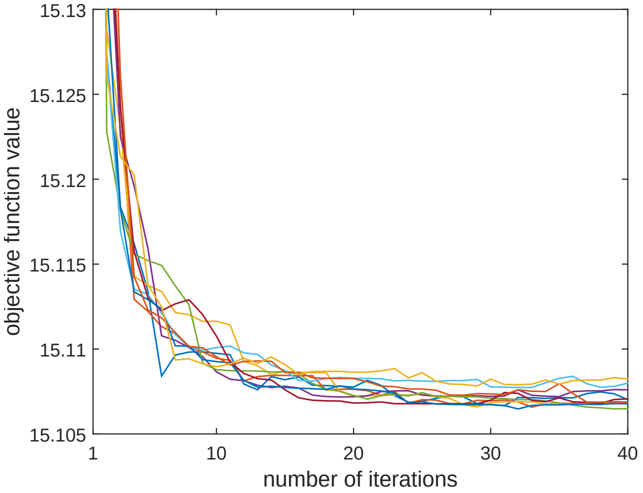


Fig. 4: Repetitive experiments show that a number of 20 to 30 iterations give a satisfying result in terms of convergence.

3) Benchmarks for Comparison: We compare the performance of the On-DS algorithm with a few typical benchmarks to evaluate the impact of base traffic prediction error, the benefit of updating the prediction in real time, and the impact of deferrable traffic's penetration level. We thus consider the followings five cases in our experiments:

(0) *Offline demand shaping w/ Off-DS algorithm.* We use "real" trace for future base traffic and use arrival information recorded from case (1) below for DAs. Applied with Off-DS algorithm, this case gives the optimal performance used as benchmark to characterize the gap of other cases.

(1) *Online demand shaping w/ On-DS algorithm.* We make prediction for both DAs' arrival and base traffic in the future. Prediction is updated at each timeslot. We run On-DS algorithm to schedule traffic.

(2) *Online demand shaping w/ exact information for base traffic and w/o exact information for DAs.* We use "real" trace for base traffic and prediction for DAs. We apply On-DS algorithm. Comparison of case (2) with case (1) shows the impact of uncertainty in base traffic.

(3) *Demand shaping w/ updating prediction of base traffic and w/ exact information for DAs.* We use DAs arrival information recorded from case (1). Instead of applying virtual deferrable traffic, we schedule traffic profiles for all the future deferrable traffic. Since the exact base traffic information is not available, we updated base traffic prediction at each timeslot. Comparison of case (3) and (1) shows the impact of uncertainty in DAs arrival prediction.

(4) *Demand shaping w/o updating prediction of base traffic and w/ exact information for DAs.* We use prediction of the base traffic at the beginning ($t = 1$) without further updating, and use arrival information recorded from case (1) for DAs. This case shows how the online algorithm benefits from updating prediction at each timeslot.

We use the metric of relative gap $G_r(d) = (V(d) - V(d^0))/V(d^0)$ to evaluate the performance, where d^0 is the

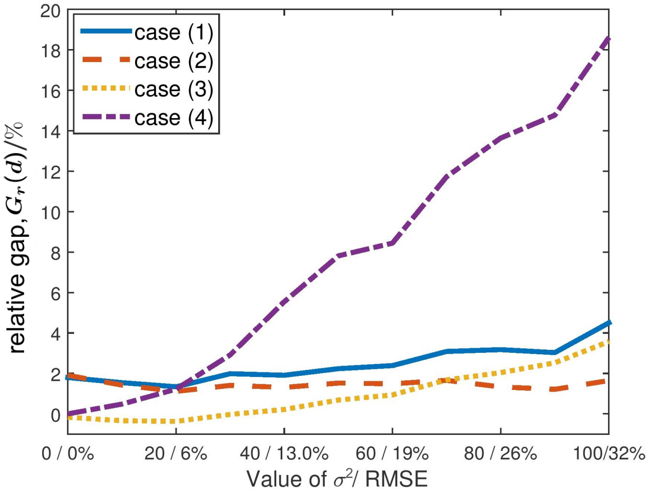


Fig. 5: Base traffic prediction error has little impact on online algorithms with updated base traffic prediction.

results obtained from case (0). Also notice that when d is calculated based on case (1), $G_r(d)$ becomes G_r^{on} in Theorem 6.

B. Experiment Results

Considering randomness in DAs' arrivals, base traffic prediction, and deciding traffic profiles for discrete DAs, we run simulation for 10 times, and take the average as the final result to present.

1) Convergence Speed: We first run a case of randomly generated 143 continuous DAs and 150 discrete DAs by Off-DS algorithm with different numbers of iterations ranging from 1 to 40 for 10 times. Because of the random process in choosing traffic profiles for discrete DAs, we observe oscillation in objective function values for each individual run. However, the oscillation has a trend of diminishing as the more iterations are implemented, with satisfying enough results generated from running 20 to 30 iterations. See Fig.4 for the results. We will implement a number of 30 iterations to make each decision for the rest of simulation.

2) Impact of Base Traffic Prediction Error: As described in Section VI-A, we can tune the variance σ^2 to emulate situations with different prediction errors in base traffic. As Fig. 5 shows, with updated prediction, case (1)'s performance is barely affected by the increasing prediction error, keeping its relative gap under 5%. This is almost as good as that of case (2) with perfect base traffic information. We can also see from the performance of case (3) the pure impact from prediction errors, while case (4) gives an example showing what happens if there is no updated prediction.

3) Impact of Penetration Level of Discrete DAs: In this case, we fix the prediction error in base traffic at $\sigma^2 = 40$ and the average number of DAs' arrival at each timeslot at $\lambda_p = 4$. We then tune the penetration level of discrete DAs from 25% to 75% with granularity of 5%. As shown in Fig. 6, the relative gap maintains relatively unaffected by the changes of discrete DAs whose penetration has increased by three times. Here, we do *not* observe a decreasing relative gap mainly because the gap is not monotonically decreasing with number of N''.

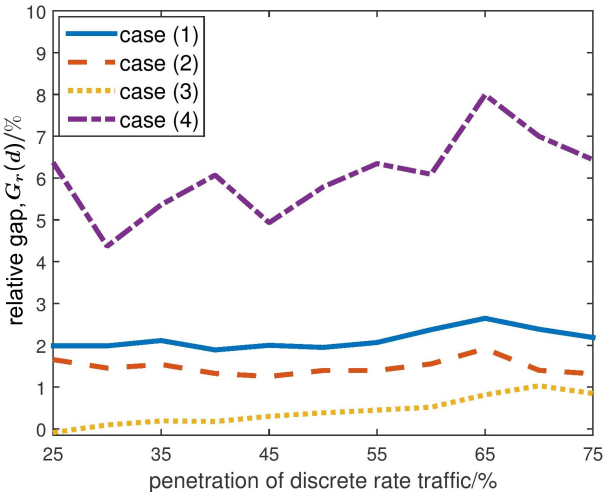


Fig. 6: Increasing penetration level of deferrable traffic does not influence the relative gap of online algorithms.

VII. Conclusion

We have formulated demand shaping in cellular networks as an optimization problem that minimizes the time variation in aggregate traffic subject to the rate and time requirements of the applications. We design a distributed and randomized offline demand shaping algorithm under complete traffic information and prove its almost surely convergence. We then consider a realistic setting with incomplete information where we can only predict future traffic to a certain degree of accuracy, and design an online demand shaping algorithm that updates the schedules of deferrable applications each time new information is available, based on solving at each timeslot an optimization problem over a shrinking horizon from the current time to the end of the day. We compare the performance of the online algorithm against the optimal offline algorithm analytically and numerically. As future work, we are investigating to integrate the incentive mechanisms such as the smart data pricing into the demand shaping algorithm design. We also plan to develop a platform to enable automatic demand shaping in cellular networks and investigate the related practical issues.

Acknowledgement

We would like to thank Seungil You for help with simulations and Lingwen Gan for careful comments.

References

[1] Airtel launches unlimited-usage night plans for calls, internet. http://businesstoday.in.intoday.in/story/airtel-night-plans-unlimited-usage-for-calls-internet/1/205272.html.

[2] AT&T still throttles "unlimited data"—even when network not congested. https://arstechnica.com/information-technology/2014/12/att-still-throttles-unlimited-data-even-when-network-not-congested/.

[3] How much bandwidth does Skype need? https://support.skype.com/en/faq/FA1417/how-much-bandwidth-does-skype-need.

[4] Netflix internet connection speed recommendations. https://help.netflix.com/en/node/306.

[5] Sprint night and weekend minutes. http://shop2.sprint.com/en/stores/popups/voice_nights_weekends_7pm_popup.shtml.

[6] T-mobile now throttling mobile hotspots when network is congested. https://arstechnica.com/information-technology/2016/10/t-mobile-now-throttling-mobile-hotspots-when-network-is-congested/.

[7] Telkom night surfer plan. http://www.telkommobile.co.za/plans/prepaid-data/60gbpromo/.

[8] Verizon nationwide for business plans. http://business.verizonwireless.com/content/b2b/en/shop-business-products/business-plans/nationwide-for-business.html.

[9] Verizon wireless to slow down users with unlimited 4G LTE plans. https://arstechnica.com/information-technology/2014/07/verizon-wireless-to-slow-down-users-with-unlimited-4g-lte-plans/.

[10] A. Balasubramanian, R. Mahajan, and A. Venkataramani. Augmenting mobile 3G using WiFi. *Proceedings of International Conference on Mobile Systems, Applications, and Services*, pages 209–222, 2010.

[11] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[12] L. Chen, L. Jiang, N. Li, and S. H. Low. Optimal demand response: Problem formulation and deterministic case. *Control and Optimization Theory for Electric Smart Grids*, 2012.

[13] M. H. Cheung, F. Hou, J. Huang, and R. Southwell. Congestion-aware distributed network selection for integrated cellular and Wi-Fi networks. *IEEE Journal on Selected Areas in Communications*, 35(6):1269 –1281, 2017.

[14] L. Gan, U. Topcu, and S. H. Low. Stochastic distributed protocol for electric vehicle charging with discrete charging rate. *Proceedings of Power and Energy Society General Meeting*, pages 1–8, 2012.

[15] L. Gan, A. Wierman, U. Topcu, N. Chen, and S. H. Low. Real-time deferrable load control: Handling the uncertainties of renewable generation. *Proceedings of International Conference on Future Energy Systems*, pages 113–124, 2013.

[16] G. R. Grimmett and D. R. Stirzaker. *Probability and Random Process*. Oxford University Press, third edition, 2001.

[17] S. Ha, S. Sen, C. Joe-Wong, Y. Im, and M. Chiang. Tube survey questions and demographics. http://www.princeton.edu/∼cjoe/TUBE_Survey.pdf, Jan 2012.

[18] S. Ha, S. Sen, C. Joe-Wong, Y Im, and M. Chiang. Tube: Time-dependent pricing for mobile data. *ACM SIGCOMM Computer Communication Review*, 42(4):247–258, 2012.

[19] M. H. Hajiesmaili, L. Deng, M. Chen, and Z. Li. Incentivizing device-to-device load balancing for cellular networks: An online auction design. *IEEE Journal on Selected Areas in Communications*, 35(2):265–279, 2017.

[20] Cisco Visual Networking Index. Global mobile data traffic forecast update, 2016–2021. 2017.

[21] Chitika Insights. Hour-by-hour examination: Smartphone, tablet, and desktop usage rates, 2013.

[22] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas. A double-auction mechanism for mobile data-offloading markets. *IEEE/ACM Transactions on Networking (TON)*, 23(5):1634–1647, 2015.

[23] L. Jiang, S. Parekh, and J. Walrand. Time-dependent network pricing and bandwidth trading. *Proceedings of IEEE Network Operations and Management Symposium Workshops*, pages 193–200, 2008.

[24] C. Joe-Wong, S. Ha, and M. Chiang. Time-dependent broadband pricing: Feasibility and benefits. *Proceedings of International Conference on Distributed Computing Systems (ICDCS)*, pages 288–298, 2011.

[25] W. H. Kwon and A.E. Pearson. A modified quadratic cost problem and feedback stabilization of a linear system. *IEEE Transactions on Automatic Control*, 22(5):838–842, Oct 1977.

[26] J. Lee, Y. Yi, S. Chong, and Y. Jin. Economics of WiFi offloading: Trading delay for cellular capacity. *IEEE Transactions on Wireless Communications*, 13(3):1540–1554, 2014.

[27] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong. Mobile data offloading: How much can WiFi deliver? *IEEE/ACM Transactions on Networking*, 21(2):536–550, 2013.

[28] N. Li. A market mechanism for electric distribution networks. *Proceedings of IEEE Annual Conference on Decision and Control (CDC)*, pages 2276–2282, 2015.

[29] N. Li, L. Chen, and S. H. Low. Optimal demand response based on utility maximization in power networks. *Proceedings of IEEE Power Engineering Society General Meeting*, July 2011.

[30] F. Mehmeti and T. Spyropoulos. Is it worth to be patient? Analysis and optimization of delayed mobile data offloading. *Proceedings of IEEE INFOCOM*, pages 2364–2372, 2014.

[31] F. Parise, M. Colombino, S. Grammatico, and J. Lygeros. Mean field constrained charging policy for large populations of plug-in electric vehicles. *Proceedings of IEEE Annual Conference on Decision and Control (CDC)*, pages 5101–5106, 2014.

[32] I. C. Paschalidis and J. N. Tsitsiklis. Congestion-dependent pricing of network services. *IEEE/ACM Transactions on Networking*, 8(2):171–184, 2000.

[33] F. A. Potra and S. J. Wright. Interior-point methods. *Journal of Computational and Applied Mathematics*, 124(1):281–302, 2000.

[34] S. Sen, C. Joe-Wong, S. Ha, and M. Chiang. Pricing data: A look at past proposals, current plans, and future trends. *arXiv preprint arXiv:1201.4197*, 2012.

[35] S. Sen, C. Joe-Wong, S. Ha, and M. Chiang. Smart data pricing (SDP): Economic solutions to network congestion. *SIGCOMM eBook on Recent Advances in Networking*, 2013.

[36] J. Tadrous, A. Eryilmaz, and H. El Gamal. Proactive resource allocation: Harnessing the diversity and multicast gains. *IEEE Transactions on Information Theory*, 59(8):4833–4854, 2013.

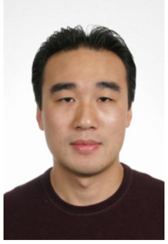
[37] L. Zhang. *Smart Data Pricing in Wireless Data Networks: An Economic Solution to Congestion*. PhD thesis, The Hong Kong Polytechnic University, 2016.

[38] S. Zhao, X. Lin, and M. Chen. Peak-minimizing online EV charging: Price-of-uncertainty and algorithm robustification. *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, pages 2335–2343, 2015.

[39] X. Zhou and L. Chen. Demand shaping in cellular networks. *Proceedings of Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 621–628, 2014.

[40] X. Zhou, E. Dall'Anese, L. Chen, and A. Simonetto. An incentive-based online optimization framework for distribution grids. *IEEE Transactions on Automatic Control*, 2017.

Xinyang Zhou received a B.S. in Engineering from Zhejiang University, China in 2012. He is currently a PhD candidate in Telecommunications at the University of Colorado at Boulder. His current research interest is in distributed optimization and control for power systems.



Lijun Chen (M'05) is an Assistant Professor of Computer Science and Telecommunications at University of Colorado at Boulder. He received a Ph.D. in Control and Dynamical Systems from California Institute of Technology in 2007. He was a co-recipient of the Best Paper Award at the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS) in 2007. His current research interests include optimization and control of networked systems, distributed optimization and control, convex relaxation and parsimonious solutions, and game theory and its engineering application. He is an editor for the IEEE Transactions on Communications and editor for the IEEE Control Systems Society Conference Editorial Board.