TUBE: Time-Dependent Pricing for Mobile Data

Sangtae Ha* Soumya Sen* Carlee Joe-Wong* Youngbin Im*,† Mung Chiang*

*Princeton University †Seoul National University

{sangtaeh, soumyas, cjoe, chiangm}@princeton.edu ybim@mmlab.snu.ac.kr

ABSTRACT

The two largest U.S. wireless ISPs have recently moved towards usage-based pricing to better manage the growing demand on their networks. Yet usage-based pricing still requires ISPs to over-provision capacity for demand at peak times of the day. Time-dependent pricing (TDP) addresses this problem by considering *when* a user consumes data, in addition to how much is used. We present the architecture, implementation, and a user trial of an end-to-end TDP system called TUBE. TUBE creates a price-based feedback control loop between an ISP and its end users. On the ISP side, it computes TDP prices so as to balance the cost of congestion during peak periods with that of offering lower prices in less congested periods. On mobile devices, it provides a graphical user interface that allows users to respond to the offered prices either by themselves or using an "autopilot" mode. We conducted a pilot TUBE trial with 50 iPhone or iPad 3G data users, who were charged according to our TDP algorithms. Our results show that TDP benefits both operators and customers, flattening the temporal fluctuation of demand while allowing users to save money by choosing the time and volume of their usage.

Categories and Subject Descriptors: C.2.3[**Computer-Communication Networks**]: Network Operations—*Network Management*

General Terms: Economics, Human Factors, Management

Keywords: Time-dependent pricing, User trial, Wireless

1. INTRODUCTION

While researchers have proposed different plans for pricing data usage for many years, wireless ISPs have traditionally used only flat-rate, usage-based, or simple day/night charging. However, the recent rapid growth in demand for data [2] is forcing them to explore new ways to match revenues to costs. Dynamic *time-dependent pricing* (TDP) is one way to do so. With TDP, an ISP can offer lower prices in less-congested periods, incentivizing users to shift their usage from peak to off-peak periods. In fact, many applications today, e.g., movie or software downloads and cloud data synchronization, have significant delay tolerances and can be deferred to low usage periods if proper incentives are provided. Other applications can tolerate shorter, but still useful, delays. These deferrals can reduce the peak traffic: our partner ISP data shows that the demand in peak hours can be ten times that in off-peak hours. Even within ten minutes, demand can vary by a factor of two. TDP leverages this traffic pattern to help ISPs reduce the cost of peak-load provisioning for their networks, while allowing users to save money by choosing the time of their usage.

Implementing such a TDP plan requires *architecting* and *prototyping* a fully functional system that enables ISPs to offer prices acceptable to both themselves and end users. Additionally, it requires developing simple and intuitive GUIs that let users view and respond to the offered prices. In this paper, we present the design, implementation, and pilot trial evaluation of the TUBE (Time-dependent Usage-based Broadband price Engineering) system for mobile data [18].

1.1 Matching Price to Cost

The proliferation of high-speed LTE, smartphones, tablets, bandwidth-hungry apps, and cloud-based services has brought about an explosive growth in wireless Internet usage. The heavy tail of this usage distribution, which largely drives ISPs' operational costs, has led to the demise of 'flat rate' unlimited data plans in countries like the U.S. [5, 24, 28]. ISPs are now pursuing measures such as tiered usage pricing, overage charges, aggressive throttling, and service discontinuation to alleviate congestion on their networks [1]. By the spring of 2012, both AT&T and Verizon Wireless had announced updates to their mobile data usage policies, effectively imposing usage-based pricing of about $10/GB [6]. But as Clark [3] observed:

> *The fundamental problem with simple usage fees is that they impose usage costs on users regardless of whether the network is congested or not.*

Moreover, usage-based fees fail to address ISPs' real problem. Heavy users typically congest the network at the same time, resulting in large demand peaks that force ISPs to over-provision network capacity and incur costs accordingly. Solving this problem requires a viable economic model and system capability for charging users by not only how much data they consume but also when they do so. This idea of 'responsive pricing' was advocated by Mackie-Mason et al. [14] as early as 1995 when the commercial Internet was still evolving:

SIGCOMM'12, August 13–17, 2012, Helsinki, Finland.

We argue that a feedback signal in the form of a variable price for network service is a workable tool to aid network operators in controlling Internet traffic. We suggest that these prices should vary dynamically based on the current utilization of network resources.

The general concept of congestion pricing has been studied in the energy, transportation, telephone, ATM, and IP network research communities for several decades. Only recently, however, has congestion-dependent pricing become realistic for mobile data. For example, Uninor in India and MTN in Uganda already offer congestion- and location-dependent pricing for voice calls [25].

1.2 Opportunities and Challenges

Time-dependent pricing for mobile data is a natural step in the transition from simple usage pricing to congestion-dependent pricing. Unlike voice calls, many mobile apps have different degrees of time sensitivity and do not require real-time data transfers or user interactivity. Time-dependent pricing can exploit these features to create effective price incentives for users to flatten their temporal demand profile by adjusting prices to user response. But TDP for mobile data also presents several new technological and social challenges:

- How can ISPs compute price incentives that they are willing to offer and users are ready to accept?
- Can we develop economic models that can be easily estimated from real data?
- What are the key system design challenges?
- How can we assess the benefits of TDP?
- Will real users respond to TDP favorably?
- How can we minimize user interaction from the client side?

Answering each of these questions requires significant effort in conducting consumer surveys, developing analytical models, building a system prototype, and finally running trials with real users. The aim and scope of such an undertaking echoes those of the Berkeley INDEX project for usage-based wireline pricing [27] and the work in [21] for voice calls. Yet the present context of wireless TDP introduces new requirements, challenges, and opportunities, including the following:

Dynamic TDP: To compute time-dependent prices, ISPs must estimate the amount of usage that will be shifted to lower-price periods as a function of the prices offered. However, this estimation should change over time: as an ISP offers time-dependent prices and observes the resulting network demand, it must adjust and improve its estimates of user reaction to the prices offered to better reflect the usage observed. These changes in user behavior estimates then prompt changes in the prices offered, forming a *feedback loop* interaction shown in Fig. 1.

User behavior estimation: Given the prices offered over a day, users will shift different amounts of different types of traffic to lower-price periods. For instance, some users may wait for five minutes but not an hour to stream movies. Similarly, a user may wait to refresh a personalized news magazine, but not to download urgent email attachments. The ISP's model of user behavior must therefore account for this heterogeneity in users' reactions to the offered prices. It must also be readily adjustable to observed changes in *aggregate* demand across users.

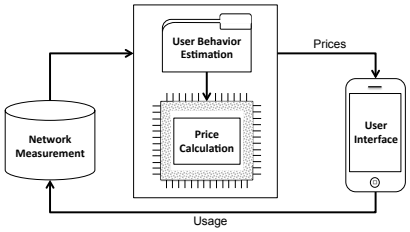


Figure 1: Feedback-loop schematic of TUBE.

User experience: When prices are published to users' mobile devices, each user optimizes her behavior to satisfy her goals, e.g., spending less than a certain monthly budget. Most users, however, will not manually optimize their usage to do so. Thus, TDP requires an "autopilot" mode that can automatically schedule usage for users. To make the system scalable to multiple users and prevent privacy violations, this autopilot should operate separately from the ISP.

1.3 TUBE Contributions

While prototyping and deploying TUBE in a trial with 3G data users, we focus on five main features:

A fully functional system for offering TDP for mobile data. Deploying TDP as a new mobile data pricing plan requires taking this idea from economic theory to a fully integrated system. We build a model for dynamic TDP that incorporates evolving user behavior, populate the model parameters from user surveys, design a supporting architecture, implement a prototype, and, finally, run a pilot trial with real users.

An architecture using feedback control. TUBE creates a feedback loop between the ISP server, which computes the prices to offer users, and the users who respond to the offered prices. The ISP offers prices on a *day-ahead* basis: at any given time, users know the prices for the next twenty-four hours. Day-ahead prices provide some certainty for users to plan ahead, while allowing ISPs to adjust prices each day according to revised user behavior estimates.[1]

User behavior models and optimized price computation. We propose an economic model of user behavior, as well as an algorithm for dynamically estimating relevant parameters from aggregate demand. The model helps predict usage for subsequent TDP periods, allowing the ISP to optimize its prices. These prices balance the cost of handling high demand relative to capacity with that of offering price discounts to users.

An user interface design. We study psychological aspects of user interaction and offer an autopilot mode for scheduling apps that minimizes "human-in-the-loop" issues. Users are thus able to optimize their usage with respect to the time-dependent prices independent of the ISP.

A realistic evaluation with real users. We recruited 50 iPhone/iPad 3G users on our university campus as trial participants. We charged participants according to TDP by deploying the TUBE prototype on our server and participant devices. Our results indicate that users indeed respond to prices: when offered monetary discounts, they will shift their demand from peak- to off-peak periods and even consume more in off-peak periods.

The overall TUBE architecture and the details of our user

[1]Of course, the prices need not be offered a full day ahead of time; TUBE allows ISPs to choose any time-window size. Pricing for sudden increases in demand due to special events can also be accommodated.

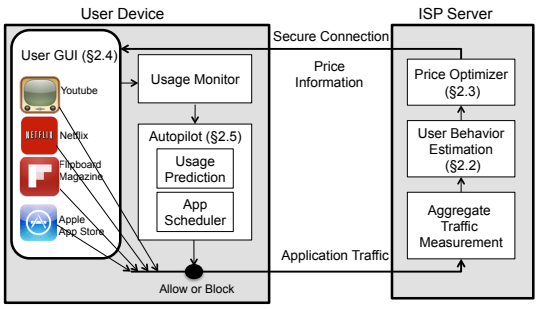


Figure 2: TUBE architecture showing the user-side and ISP-side components and interaction.

behavior modeling and price computation are described in Section 2. Section 3 next discusses our prototype implementation on the server and user devices, and Section 4 discusses the setup, logistics, and results of our pilot trial. Section 5 points out limitations of our work and future directions, while Section 6 discusses related work in this field. Finally, Section 7 concludes the paper.

2. ARCHITECTURE AND DESIGN

The TUBE architecture consists of two main components, as shown in Fig. 2. One component resides on the ISP server and computes the prices offered to users, while the other resides on user devices.

The ISP-side component of Fig. 2 measures individual usage and determines the prices offered to the end users.[2] These prices optimize the ISP's profit given its evolving estimates of user behavior.

At regular intervals, the user device pulls the price information over a secure connection and displays the prices computed on the ISP server. The usage monitor in the device measures the timing and volume of each application's usage. If activated by the users, the autopilot mode uses these measurements to schedule applications on their behalf.

In this section, we first discuss our guiding design principles (Section 2.1). Next, we consider two important modules on the ISP server: how TUBE estimates user behavior (Section 2.2) and how it optimizes the TDP prices (Section 2.3). Finally, we discuss our GUI designs (Section 2.4) and an autopilot algorithm (Section 2.5) that runs on user devices.

2.1 Design Guidelines

Our design choices stem from the following principles:

Separating functionality: The ISP server estimates users' reactions to prices and solves a large-scale nonlinear optimization to compute the prices offered to users. Since user devices have limited computing power, we run this price calculation on a central server. To allow users to view the prices and automatically respond by scheduling their usage, we implement a user interface on their individual devices.

Scaling up the system: To maintain scalability, our behavior estimation algorithm requires only aggregate, and not individual, usage data. We formulate the price calculation as a convex optimization problem, which can be rapidly solved for many TDP periods.

Protecting user privacy: TUBE requires neither monitoring of users' application source/destination addresses, nor any Deep Packet Inspection (DPI). The only data exchanged between a user device and ISP servers are the prices and usage in each period, which are secured with TLS/SSL connections.

Empowering user control: The user interface displays the prices offered and corresponding device usage, allowing users to educate themselves about their data consumption and schedule their usage according to the prices offered. To facilitate this educational component, we use simple and intuitive GUI designs.

[2]Although the behavior estimation requires only aggregate data, ISPs can of course keep track of individual usage data in order to calculate users' monthly bills.

2.2 Estimating User Behavior

In order to set time-dependent prices, ISPs monitor users' traffic patterns, i.e., the volume of traffic at different times, with and without TDP. This data is then used to estimate users' willingness to shift their traffic in exchange for a monetary discount. The estimates are used to calculate the time-dependent prices for the next day, linking the prices and user behavior in the system's feedback loop (Figs. 1 and 2).

In this section, we first discuss a parameterized model of user behavior and use consumer survey results to initialize the parameter values. We then provide an algorithm to update these user behavior estimates and evaluate its effectiveness on simulated data.

2.2.1 *Modeling Delay Tolerances*

Users' willingness to wait for data usage depends on the type of session under consideration: for instance, iTunes downloads can often be more readily delayed than streaming YouTube videos. Thus, we view each user as a set of *application sessions*, e.g., streaming, browsing, and file transfer sessions. Sessions are assumed to have a fixed minimum bandwidth requirement, which is particularly appropriate for streaming sessions.

In general, users' willingness to defer any given application session depends on two factors: the monetary reward for deferring the session and the time for which the session is deferred. For instance, users may wait for 1 hour to watch a video in exchange for \$2, but may not wait if offered only \$1. At the same time, users may not wait more than one hour, even if offered the \$2 reward.

In order to quantify users' willingness to defer application sessions, we introduce the concept of *waiting functions*. These functions give the probability that users will defer an application session for a given amount of time τ in exchange for a discount d, e.g., d \$/GB from some baseline metered price. Since different users and applications may have distinct waiting functions, we choose a parameterized form of the waiting functions, with different parameters corresponding to different levels of user patience. These parameters thus quantify the various price-delay tradeoffs corresponding to different users and application sessions.

The chosen form of the waiting functions should be decreasing in the time deferred, τ, and concave and increasing in the discount offered d: users will be less likely to defer as the time deferred increases but will be more likely to defer if offered a larger monetary reward. While many functional forms are reasonable, we choose the simple form

$$w_\rho(d,\tau) = \frac{d}{\lambda_\rho(\tau+1)^\rho}, \qquad (1)$$

where ρ is a parameter measuring patience, the *patience index*, and λ_ρ is an appropriate normalization constant. A

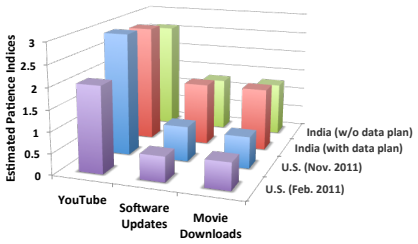


Figure 3: Patience indices from survey results.

larger ρ indicates less patience, while $\rho = 0$ indicates indifference to the time deferred τ: as τ increases, willingness to wait w_ρ drops off faster for larger ρ. The normalization constant is chosen so that the $w_\rho(d, \tau)$ could represent the probability that users will defer for τ amount of time, given the discount d. Thus, we generally choose $\lambda_\rho = \sum_{\tau=1}^{n-1} w_\rho(D, \tau)$, where D denotes the maximum possible discount offered in each period and $w_\rho(D, \tau)$ is the unnormalized value of (1).[3] The summation of τ from 1 to $n-1$ represents the possible times deferred (1 period, 2 periods, etc.) up to one day, or n periods. We expect that users will not have significant incentive to delay their traffic for more than one day.

2.2.2 *Initializing Patience Indices*

In this section, we give reasonable estimates of the patience indices ρ used in (1) to model users' willingness to delay their data consumption. In practice, such estimates may be achieved with a calibration period of pilot trials in which the ISP offers a wide variety of prices to users and observes their responses. In the present work, we take a different approach with initial market surveys in the U.S. and India. We conducted two U.S. surveys, one in February 2011 and one in November 2011, each with 130 participants from 25 states. The India survey included 546 participants from 5 large cities.

In each survey, we asked the respondents whether they would delay a given application type, e.g., YouTube videos or software updates, for specified time intervals. Participants were told that waiting would reduce their bill by two-thirds in the first U.S. and India surveys and by one-third in the second U.S. survey. The time interval choices ranged from 5 minutes (e.g., for YouTube videos) to 48 hours (e.g., for software updates). The survey questions are given in [8].

We first use the survey responses to find the fraction of users willing to wait for each of these time intervals. Given these fractions, we then compute their discrete derivative with respect to time (i.e., the differences between the fractions divided by the interval duration) to find the waiting function values for each traffic class at the survey-specified discount. Figure 3 shows the resulting patience indices for three different traffic classes. As expected, in all three surveys, participants are much more likely to wait for software updates or movie downloads than YouTube streaming, reflecting streaming's more "immediate" utility to users.

2.2.3 *Updating Patience Indices*

We now give an algorithm for ISPs to update their estimates of (1)'s user patience indices for different applications. We assume that the ISP has several days of aggregate (cross-user and cross-application) usage data and the corresponding prices offered in each period.

The main idea of our algorithm is to compare the observed usage data with a baseline usage trace, taken for time-independent prices (TIP). We assume that changes between the baseline usage and observed usage with TDP are due to the time-dependent prices offered. These changes can then be expressed in terms of the waiting function parameters and volume of different application types; we can use standard curve-fitting methods to estimate the optimal waiting function parameters. The following discussion details this procedure, which is summarized in Algorithm 1.

Algorithm 1: Waiting function estimation

Data: Prices offered and TDP traffic pattern.
Result: Estimated waiting functions.
Compute the S_i, to obtain n linear equations for the A_{ik};
Solve for $n-2$ of the A_{ik}, such that for each period j, at least one A_{jk} is not solved for;
Substitute these expressions back into the original equations for S_i, so that only one equation, linear in the A_{ik}, remains;
// This equation is a function of parameters μ_j and ρ_j and the offered discounts.
Use the TIP and TDP data for this function to estimate (e.g., with nonlinear least-squares) all the μ_j and ρ_j parameters it contains;
// These μ_j and ρ_j values give the waiting functions.

Since the baseline traces have only aggregate usage data, the ISP does not know the usage for each application. Yet for our purposes, grouping sessions by application type has an important disadvantage: depending on the user, a given application may correspond to different patience indices. We therefore group applications by *traffic class*, defined as a group of sessions with the same patience index, rather than application type. After choosing a number of traffic classes, we must estimate both the patience index and the fraction of traffic corresponding to each traffic class.

Our estimation algorithm relies on calculating the expected amount of traffic deferred from a given period i to another period k in terms of the waiting function parameters.[4] Thus, the amount of traffic in period i without any traffic shifting is simply the average amount of traffic in period i under TIP. Given a set of discounts d_i, $i = 1, 2, \ldots, n$, offered over one day, the expected amount of traffic deferred from period i to period $k \neq i$ is then

$$A_{ik} = Y_i \sum_{j=1}^{m} \mu_j w_{\rho_j}\left(d_k, |k-i|_n\right), \tag{2}$$

where Y_i is the TIP usage in period i, and $|k-i|_n$ is understood to be modulo n, representing the time difference between period i and the nearest period k after period i. If $k < i$, period k will occur on the day after period i. There are m traffic classes, with the jth traffic class having patience index ρ_j and taking up a proportion μ_j of the traffic. Denoting by S_i the difference between the TIP and TDP traffic in period i, we see that each $S_i = \sum_{k \neq i} A_{ik} - A_{ki}$.

Each of the n S_i values is a linear combination of the $n(n-1)/2$ A_{ik} variables. One equation may be eliminated, since we assume the sum of the S_i is zero (no traffic is lost with TDP). We can thus reduce this system of $n-1$ linear equations for the S_i to one equation, by solving for $n-2$ of the A_{ik} in terms of other A_{ik} variables. The ISP can then estimate the parameters μ_j and ρ_j using Algorithm 1.

[3]For instance, D may be the baseline metered price.

[4]For simplicity, we assume that the average traffic volume over each day remains the same when TDP is introduced, i.e., that no traffic is lost due to TDP.

Period	Actual			Estimated			Error
	ρ_1	ρ_2	μ_1	ρ_1	ρ_2	μ_1	(%)
1	1	2	0.17	1.03	2.48	0.46	11.8
2	1	2.33	0.5	1.02	2.49	0.45	9.0
3	1	2.67	0.83	0.90	2.15	0.71	0.5

Table 1: Actual and estimated parameter values in a simulation of waiting function estimation.

Since baseline TIP demand may change over time, we use the following procedure to adjust the baseline. The ISP estimates the waiting functions using TDP data from several days, e.g. one month, and uses these estimates to solve for the demand under TIP, Y_i, in each period i. The n equations (2) are linear in the Y_i, and all other variables are known upon choosing a set of discounts offered and traffic observed.[5] We then estimate the Y_i values.

To illustrate Algorithm 1, we consider a simple example with two traffic classes and three periods. Actual parameter values are given in Table 1. Our simulation takes a set of given discounts and computes the traffic if waiting functions are perfectly followed, adding noise to these results. Table 1 shows the μ_j and ρ_j values estimated by nonlinear least squares. The maximum percent difference between actual and estimated waiting function values $\mu_1 w_{\rho_1} + (1-\mu_1) w_{\rho_2}$, measured on a 1000-point grid of discounts and times deferred, remains small at under 12%.

2.3 Optimizing Prices

We now describe how to use the estimated waiting functions to calculate time-dependent prices over the next day. An ISP wishes to set prices that balance two types of costs: that of exceeding the maximum capacity, and that of offering discounts to users in less-congested time periods. We take the "maximum capacity" to mean the maximum amount of traffic that can be handled by the network with an acceptable amount of congestion. Thus, demand may "exceed capacity" in the sense that over a certain capacity, user response time due to congestion becomes unacceptably high.

Suppose that there are n periods in a day, each lasting one unit (e.g., hour or half-hour) of time. We assume that the ISP's network has a bottleneck link of capacity C_i in period i, defined as the link capacity less any background traffic or excess capacity "cushion." The cost of exceeding capacity is assumed to be piecewise-linear and convex; we denote the cost in period i as $g(y_i - C_i)$, where y_i is the predicted usage with TDP discounts d_i in period i. For ease of notation, $j \in i$ indexes all traffic classes j in period i. Moreover, the time between periods i and k is given by $|i-k|_n$ as in (2).

We can now calculate the costs of exceeding capacity Γ_1 and offering discounts Γ_2 as

$$\Gamma_1 = \sum_{i=1}^{n} g\left(Y_i \left(1 - \sum_{j \in i} \mu_j \sum_{k \neq i}^{n} w_j\left(d_k, |k-i|_n\right) \right) + \sum_{k=1, k\neq i}^{n} Y_k \sum_{j \in k} \mu_j w_j\left(d_i, |i-k|_n\right) - C_i \right) \quad (3)$$

$$\Gamma_2 = \sum_{i=1}^{n} d_i \left(\sum_{k \neq i}^{n} Y_k \sum_{j \in k} \mu_j w_j\left(d_i, |i-k|_n\right) \right). \quad (4)$$

[5]Noise in the data results in different Y_i for different sets of discounts; the ISP can use an average of these Y_i.

Algorithm 2: Price determination.

Data: Estimated waiting functions.
Result: Optimized TDP prices.
Start with a set of discounts for the next n periods, determined with initial waiting function estimates;
while *TDP is offered* **do**
  for $k = 1 \rightarrow n$ **do**
    Choose the discount for the nth period after period k so as to minimize $\Gamma_1 + \Gamma_2$ in (3-4).
    if $k == n$ **then**
      Run the waiting function estimation (Algorithm 1) to find updated waiting function parameters.

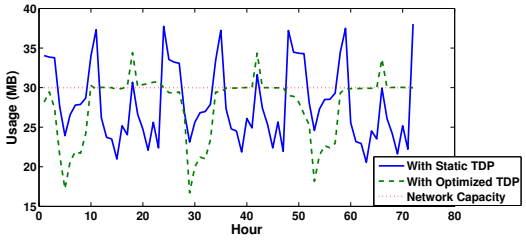


Figure 4: Static TDP vs. dynamic TDP.

The ISP attempts to minimize $\Gamma_1 + \Gamma_2$ with respect to the discount variables d_i, $i = 1, 2, \ldots, n$ [10]. For our trial, TUBE offers *day-ahead* pricing: when TDP is first introduced, the ISP publishes a full day of time-dependent prices to users. After each subsequent period, a new price for the period one day ahead of the current one is published. We thus have the online price determination in Algorithm 2.

To test the efficacy of TUBE's feedback loop, we compare simulated results of our feedback loop with results with static TDP, i.e., time-dependent prices that do not change from day to day. Figure 4 shows the results over three days. We see that the peak periods under static TDP are about 8 MB above the maximum capacity, while those under dynamic TDP are only 4 MB above the maximum capacity. The ISP underestimates users' reaction to TDP in setting the static prices; thus, many users shift to lower-price periods, resulting in a new peak period. Dynamic TDP, however, adjusts to this underestimation by correcting the prices offered, thus flattening out the peak usage.

2.4 Graphical User Interface

Users react to the prices offered through our GUIs, which are designed to be simple and intuitive. We include the following components in our design:

Price display: Users check the prices for the next 24 hours on the GUI home screen. Each price is color-coded by its discount rate, e.g., red ($< 10\%$), orange ($10 - 19\%$), yellow ($20 - 29\%$), green ($\geq 30\%$). Since users likely will not open our application to check each period's price, we also embed a colored price indicator on the status bar at the top right corner of the device screen.

Self-education: Users can view their top 5 bandwidth consuming applications to better understand their mobile data usage. The device stores usage and price history for the last three months so that the users can look up their past usage and prices by day, week, and month. The push notifier running on the ISP server informs users when they consume significant amounts of data in high-price periods.

User control: The device learns each application's usage pattern over time, enabling it to schedule applications in the autopilot mode so as to keep users from exceeding their

specified budget. Users can also override the scheduling decisions computed by the autopilot algorithm (e.g., parental control on certain applications).

2.5 Autopilot Mode

The autopilot algorithm schedules the top five applications by usage, identified from the usage monitor in Fig. 2, to keep users below their specified monthly budget. The user ranks these applications by delay tolerance, and we provide default rankings if the user does not provide them. We denote the delay tolerances by ν_j, where $j = 1, \ldots, 5$ indexes the applications and a larger ν_j indicates a higher delay tolerance. For privacy reasons, these delay tolerances, as well as usage statistics for each application, are *not* shared with the ISP server. The behavior estimation (Algorithm 1) neither requires nor utilizes this data.

Usage Prediction: To schedule application usage, we first predict usage in each period using triple exponential smoothing (TES), which incorporates *trend*, *periodic,* and *smoothing* factors into the next day's usage prediction. We use a variation of the traditional TES equations [15, 23] to incorporate periodicity over one day and over one week.

Application Scheduling: The full scheduling algorithm is presented in Algorithm 3 and outlined below. At the beginning of each day, we calculate a daily budget from the amount of money the user has left to spend, divided by the number of days left in the month. For flexibility, we scale up the budget by a factor that increases with the number of days remaining in the month: even if the user consumes a large amount of data on the current day, she can remain within the monthly budget by reducing usage in the remaining days. This scaling can be, for example, exponential. We then compare the daily budget to the expected amount spent over one day, as calculated from the predicted usage. If the expected amount spent is larger than the daily budget, we use Algorithm 3 to schedule usage so that users are predicted to remain within their daily budget. The algorithm begins with the most delay tolerant app, and defers it from some period k to period $k+1$. Here k is chosen so as to save the most money. If the user is still predicted to exceed her budget, the same app is deferred from another period k' to period $k'+1$.

If the user can no longer save money by delaying by one period, the algorithm checks whether the user would prefer to delay sessions for this application by two periods, or would prefer deferring the application of next-highest delay tolerance by one period. It continues in this fashion until the expected amount spent is lower than the daily budget. On any given day, the user is prohibited from exceeding the weekly budget, which is calculated analogously to the daily budget but over one week's duration.

Algorithm 3: Autopilot app. scheduling algorithm

Data: Prices offered and predicted usage.
Result: Usage schedule.
Initialize $h = 1$;
// h is the number of periods deferred.
Calculate predicted amount spent using the prices p_i and predicted usage u_i in each period i;
while *projected spending exceeds the daily budget* **do**
  // Choose the app k with highest delay tolerance.
  $k \leftarrow \text{argmax}_i \left\{ \nu_i : \exists \sum_{q=0}^{h-1} u_{i,j+q} > 0 \text{ s.t. } p_{j+h} < p_j \right\}$;
  // Choose the period l which will save the most money.
  $l = \text{argmax}_j \left\{ \sum_{q=0}^{h-1} u_{k,j+q} : p_{j+h} < p_j \right\}$;
  Block app k from period l to $l+h-1$, inclusive;
  // Users can choose whether or not a notification is sent at the beginning of period $l+h$ to say that the app is no longer blocked.
  Update the projected usage values;
  $h \leftarrow h+1$;
  Recalculate the predicted amount spent;
  if *app k does not have the lowest delay tolerance* **then**
    if *the user prefers to defer the app with next-highest delay tolerance by 1 period rather than defer app k by $h+1$ periods* **then**
      $h \leftarrow 1$;
      $\nu_k \leftarrow 0$;
  The user will exceed the budget under all allowable scheduling constraints;

3. SYSTEM IMPLEMENTATION

In this section, we discuss the implementation details of TUBEOpt and TUBEApp, the two main components of the TUBE system. Figure 5 shows the modules in both TUBEOpt, which resides on the ISP server, and TUBEApp, which resides on user devices.

3.1 TUBEOpt

We implemented TUBEOpt on a Linux system with an Intel Xeon 2.0 GHz CPU and 8GB of RAM. It provides a web-based API so that any device supporting web can exchange data with the server. Figure 5a shows TUBEOpt's component diagram, including the price optimization components discussed in Sections 2.2 and 2.3; the shaded blocks are part of our current implementation. All of the blocks represent dynamic modules and can be reloaded on the fly.

In the following discussion, we detail TUBEOpt's scalability of usage monitoring and the computational overhead for its user behavior estimation and price computation.

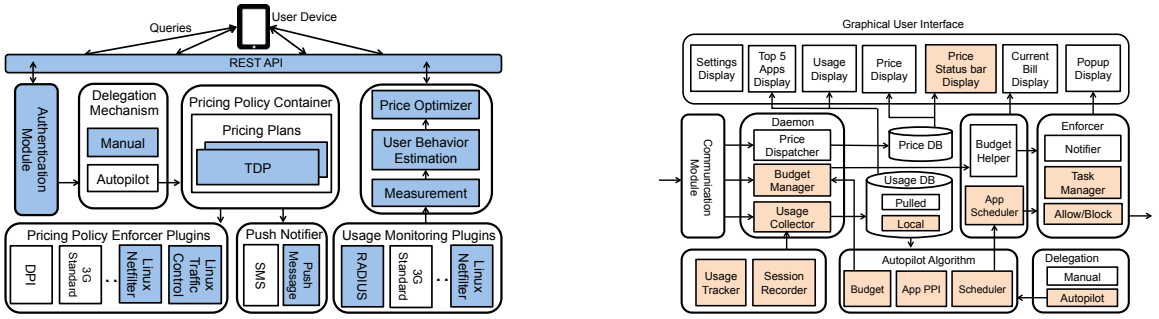
3.1.1 Improving Scalability of Usage Monitoring

To measure individual usage, we assigned a unique IP address to each user and created a `Netfilter` rule. When TUBEOpt records the usage, it retrieves the byte and packet counts from each rule. Unfortunately, this approach scales linearly with the number of users; each user requires one rule, and the computational cost increases linearly with the number of rules. While `ipset` can improve performance by combining multiple rules into one hash table, its use here is limited, as usage with `ipset` is tracked for the hash table, not for the individual rules. To improve scalability, we therefore implemented a separate kernel module that hooks the `LOCAL_IN` of the IPSEC/VPN interface (`ipsec0`). It creates a hash table and records the usage for each IP address, requiring only O(1) running time.

3.1.2 Computational Overhead

We next examine the computational overhead of using Matlab and Python to estimate user behavior and compute the optimized TDP prices. TUBE requires the runtime of these codes to be relatively short, since each period's price computation should finish before the next TDP period, at which time the server advertises the newly computed price. Therefore, we evaluate the overhead of these two computations as we increase the optimization complexity.

We measure the computational overhead (the total runtime in Matlab) as we increase the number of periods from 12 to 144 (2 hour to 10 minute periods). Table 2 shows



(a) TUBEOpt design (b) TUBEApp design

Figure 5: Building blocks of TUBE: (a) TUBEOpt on the server side and (b) TUBEApp on the mobile device. The shaded blocks in (a) are TDP-specific modules, while those in (b) require system-level modification.

Number of Periods	12	24	48	96	144
Behavior Estimation	12.76	200.0	959.6	1967	15040
Price Calculation	1.67	1.69	1.70	1.81	1.84

Table 2: Runtime of the behavior estimation and price calculation in seconds.

	Number of Application Types		
Number of Periods	2	4	8
12	0.21	12.99	21.52
24	3.33	47.08	75.47
48	15.99	197.22	215.42

Table 3: Runtime of the behavior estimation (mins).

the measured running time of the behavior estimation and price calculation. Even with 144 periods, the price calculation is quite fast (1.84 seconds); the estimation algorithm performs adequately (4.2 hours), as it runs only once a day for day-ahead TDP.

We also measure the effect of adding multiple traffic classes to the behavior estimation algorithm. Table 3 shows the algorithm running time on our Intel Xeon server. The computation with 48 periods and 8 application types still takes less than 4 hours (215.42 minutes), which is more than fast enough, as the estimation runs once a day. Our estimation uses one month of simulated data, which was generated by perturbing the usage predicted from given waiting functions by up to 50%. The running times were averaged across five computations with random data and starting points. With more powerful hardware and optimized code, significant further acceleration can be achieved.

3.2 TUBEApp

We implemented TUBEApp on the iOS, Android, and Windows platforms, although all trial participants were iOS users due to these devices' popularity on our campus. We therefore focus our discussion on the iOS implementation. We consider both the manual and autopilot modes, as shown in Fig. 5b's design and Fig. 6's GUI screenshots. Due to the iOS platform's closed nature, implementing the shaded blocks in Fig. 5b, e.g., monitoring each application's usage, requires jailbreaking the devices. We demo our app in [26].

3.2.1 OS Limitations

The Windows, Android, and iOS platforms each have various limitations; the iOS platform offers the most restrictions. While all platforms support showing the prices offered and the device's aggregate data usage, TUBEApp's autopilot mode additionally requires 1) measuring the volume of each application's usage, 2) displaying the price for the current period on a status bar, and 3) allowing and blocking bandwidth for individual applications. These features heavily depend on the openness of the platform.

Type	Status bar	App usage	Daemon support	Code size (# lines)
iPhone	No	No	Partial	25K
Android	Yes	Yes	Yes	5.4K
Windows	Yes	Yes	Yes	5.3K

Table 4: TUBEApp on different platforms.

Table 4 shows the TUBEApp features supported on each platform before hacking the device, as well as the code size for implementing the full TUBEApp. In particular, for the iOS platform, we hook several internal functions to track the usage per application, run a daemon process to dispatch and show TDP prices, and block applications. The iOS implementation thus requires 25K lines of code, while the Android and Windows implementations need only 5.4K and 5.3K lines, respectively.

3.2.2 Enhancing the User Experience

The autopilot mode minimizes user interactions by estimating device usage patterns and scheduling applications for the user. To inform users of the autopilot actions, we send pop-up notifications when usage is blocked, as shown in Fig. 7's iPhone screenshots. The warning and blocking pop-ups are displayed when the user's usage reaches the expected daily and weekly budgets (Fig. 7a and b, respectively).

To ensure that the autopilot's implementation is practical, we measure its energy usage. TUBEApp with autopilot running consumes only 4% more battery power than the device without our TUBEApp installed, indicating that autopilot does not drain too much power.

4. TRIAL DESIGN AND RESULTS

We conducted a small scale pilot trial of the TUBE system at Princeton University from May 2011 to January 2012. This section provides an overview of the trial goals, setup, and limitations, followed by a discussion of some key conclusions drawn from the trial data.

4.1 Goals and Structure

The goals of our trial are to demonstrate the TUBE system's functionality and benefits, and to provide an initial verification of TUBE's deployment feasibility with real users. Throughout the trial, we effectively acted as a resale ISP, paying the participants' regular 3G data bills to AT&T while

(a) Price display (b) Price and usage (c) Top 5 applications (d) App delay index (e) App scheduling

Figure 6: Screenshots of TUBEApp on iPhone. iPhone users can (a) check the prices for next 24 hours, (b) learn from price and usage history, (c) identify top 5 apps by bandwidth usage, (d) modify each app's delay tolerance, and (e) check when their apps are scheduled during the day and manually override the results.

charging them according to our TUBE algorithms. To assess the benefits of TDP, we divide the trial into two phases: first, we simply monitor the usage patterns without TDP (i.e., collect TIP data). We then offer TDP and study its impact in the second phase of the trial. The following aspects of the trial are addressed in this section:

Baseline Traffic Statistics: Section 4.4.1 reports on three months of TIP usage statistics from our trial participants. We ask if the sample population has a representative mix of heavy and light users and bandwidth-hungry applications, so as to realistically assess the benefits of TDP.

Price Sensitivity: In Section 4.4.2, we examine users' price sensitivity to static TDP patterns: if we offer low and high price periods alternately, will users defer their traffic to use more in the lower-price period? This question tests TDP's basic premise that users will delay their traffic in exchange for a monetary discount.

Effectiveness of GUI Design: Section 4.4.3 analyzes the effectiveness of displaying numerical values versus color codes (red: high, orange: medium, green: low) to indicate TDP prices on the user device.

Benefits of Optimized TDP: Finally, Section 4.4.4 studies whether TUBE's optimized prices benefit ISPs in reducing peak-to-average ratios of network usage.

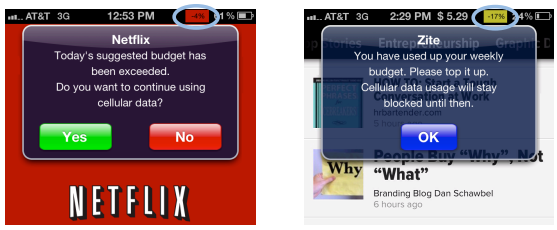
4.2 Trial Setup

We recruited 50 users (27 iPhones and 23 iPads) of AT&T's 3G Corporate Data Plan as our trial participants. They were faculty and staff from 14 academic and administrative divisions. During the trial, we acted as a resale ISP, charging participants after every billing cycle according to TUBE's TDP. We excluded measurements from development devices to avoid bias.

To record participants' usage, we separated their 3G traffic from that of other AT&T customers using an Access Point Name (APN) setup, which tunneled the participant's 3G traffic from the AT&T core to the TUBE servers in our lab (Fig. 8). Participants installed and used the TUBEApp on their iOS devices. WiFi usage, voice calls, and SMS were not included in the trial traffic as they are not 3G data.

4.3 Trial Limitations

We were limited by logistics to recruiting only AT&T data plan users with iOS devices, out of which 16 were jailbroken (JB) and 34 were non-jailbroken (non-JB) devices. The



(a) **Warning** (b) **Blocking**

Figure 7: Screenshots of auto-pilot in action on the iPhone. The warning (a) is displayed when the daily budget is reached, and usage is blocked in (b) when the user reaches the weekly budget.

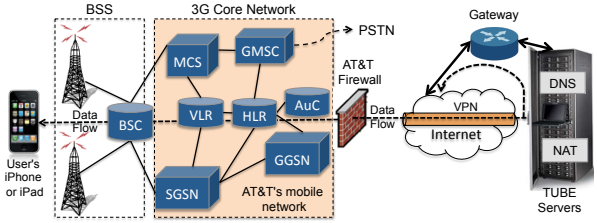


Figure 8: TUBE APN Setup. User traffic is routed through the SGSN and GGSN.

non-JB devices gave us less flexibility in experimentation, and hence ran a TUBEApp with limited features. In particular, only users with JB devices could see the current price/discount directly from the home screen's status bar (circled in Fig. 7's screenshots) without manually launching the TUBEApp, and the autopilot algorithm only runs on JB devices. On non-JB devices, we used push notifications to alert participants during high-price periods.

Since our trial included only 50 participants, even peak traffic from trial users did not congest our lab's access link. To demonstrate TDP's benefits, we simulated congestion conditions by logically scaling up the traffic volume in TUBE-Opt's price computation.[6]

4.4 Results and Evaluation

Following Section 4.1's outline, we now present the trial results. In many cases, we use *Wilcoxon's signed rank test* [12] against the null hypothesis that a set of values is sym-

[6]If users had experienced real congestion due to this scaled traffic, we expect they would have been even more willing to delay their traffic to off-peak periods.

metrically distributed with mean zero. In our case, we apply the test to the difference between the changes in usage in high- and low-price periods. A higher probability of a symmetric distribution then indicates a lack of response to the price signals, as the expected change in usage is likely the same for both high and low prices.

4.4.1 Baseline Traffic Statistics

Question: *Do our participants include both heavy and light bandwidth users? Which applications use the most data?*

Method: We measured usage for both iPad and iPhone users from July to September and used `tcpdump` to record application-specific traffic.

Results: Our participants are a mix of light- and heavy-bandwidth users. Video streaming applications accounted for most of the traffic, corroborating the reported trends of growing demand for mobile video.

Figure 9a shows the CDF of total traffic per user for uploads and downloads from July to September 2011. While 90% of the users uploaded less than 0.5GB, some users had large download volumes: 20% of users consumed 2.1 – 5.3 GB over three months.

Figure 9b shows the distribution of total traffic by application type for the three month period, normalized with respect to the number of iPhone, JB, and non-JB iPad users. Not surprisingly, iPads show a higher usage than iPhones for most application types, and a large part of the mobile traffic for all device types comes from movie streaming.

4.4.2 Price Sensitivity

Question: *Do users wait to use mobile data in return for a monetary discount*?

Method: We conducted a three week experiment on iPad and iPhone trial participants in October 2011, in which we offered a basic TDP pattern of consecutive high, high, and low price periods, repeated throughout the day. The high-price periods offered approximately a 10% discount, while the low-price periods offered a 40% discount on the baseline price of $10/GB. If monetary incentives do induce usage deferrals, we expect that average usage should decrease in high-price and increase in low-price periods.

To measure users' response to prices, we sent messages at ten minute intervals during *high-price* periods if the user exceeded 2 MB of usage in the previous ten minutes. We first analyzed the data for each user by calculating the percentage change in usage for each one-hour time period when compared to the mean usage in that same period before TDP (i.e., under TIP pricing). We then weighted these percent changes by the proportion of TDP usage in that period to account for diurnal variations. This gives the weighted average percent change in usage under TDP for high- and low-price periods.

Results: We found that *users did shift their traffic* from high- to low-price periods under TDP. For most users, the average usage decreased in high-price periods relative to the changes in low-price periods.

Figure 10 shows the weighted average percent change in usage for iPad users for high- and low-price periods. The reference line indicates an equal change in both types of periods. Each dot on the scatter plot represents values for an individual user, and its size is proportional to the user's TDP usage volume. With the given static TDP pattern, usage increased more in low-price periods relative to high-price periods for almost all users. Interestingly, about half of the users decreased their overall usage in both high- and low-price periods, while the other half increased their usage in both periods.

We further verify these results by using Wilcoxon's test on the differences between each user's percent change in high- and low-price periods. We find only a 0.56% probability that the null hypothesis is valid, indicating that the users' observed responses are statistically significant. A similar plot may be observed for the iPhone users.

The overall iPhone usage changed by -11.3% in high-price and -5.7% in low price periods, while overall iPad usage changed by -10.1% in high-price and 15.7% in low-price periods. Thus, iPad users generally decreased their usage in high-price periods and increased it in low-price periods. The overall decrease in iPhone usage is likely due to limited user notification and display options on non-JB iPhones. However, the greater usage decrease in high- relative to low-price periods indicates that iPhone users attempted to use less in high-price periods.

Next, we examine the effect of the number of notification messages sent to users on their usage in high-price periods. Multiple consecutive notifications were sent to a user only if usage in each preceding 10 minute interval exceeded 2 MB. We examine the percent change in usage in the ten minute span before and after each notification. Figure 11 shows the CDF of the percent change in usage due to a first, second, etc. notification. About 80–90% of iPad and iPhone users did not increase their usage after the first notification, indicating that notifications can effectively reduce peak usage. For all subsequent notifications, about 60–80% of the active users responded by decreasing their usage.

4.4.3 Effectiveness of User Interface Design

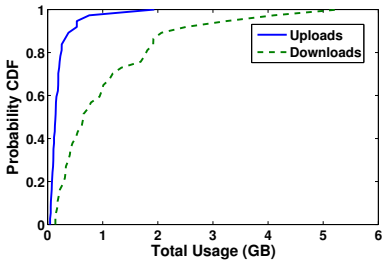
Question: *Do users respond more to the numerical values of TDP prices or to the color of the price indicator bar on the home screen?*

Method: In December 2011, we installed a price indicator bar on the home screen of all JB devices. The indicator displays the numerical value of the price discounts available in the current period and changes its color according to these discounts. It is green for discounts over 30%, orange for 10–29% discounts, and red for discounts below 10%.

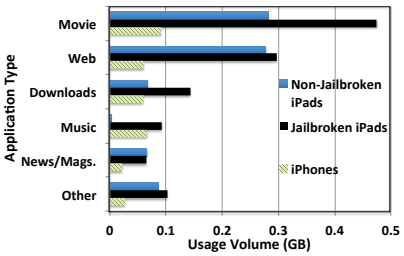
Our experiment had two stages: in the first stage, we offered discounts of approximately 40% every third period of the day, starting with a 40% discount at midnight. The other periods offered discounts of about 10%. After two weeks of following this pattern, we began the second stage, repeating the pattern of a 9% discount at midnight, followed by 28%, 30%, 28%, 9%, and 30% discounts.

We compare three types of periods to assess the effect of the indicator color and numerical discount: hours deemed as Type 1 periods offered a 10% discount in the first stage of the experiment and 28% discount in the second stage; the indicator remained orange despite this increase in the discount. Type 2 periods offered a 10% (orange) discount in the first stage and 30% (green) discount in the second stage, while Type 3 periods offered a 10% discount in the first and 9% discount in the second stage of the experiment (the indicator is orange in both periods). Table 5 summarizes the combinations of discounts and colors used in the two stages that characterize each type of period.

We calculated the percent changes in usage for each period



(a) Total traffic for each user.



(b) Total usage by app type.

Figure 9: Usage statistics from July - Sept. 2011.

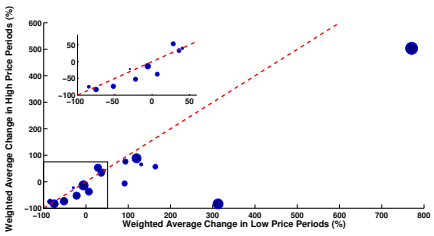
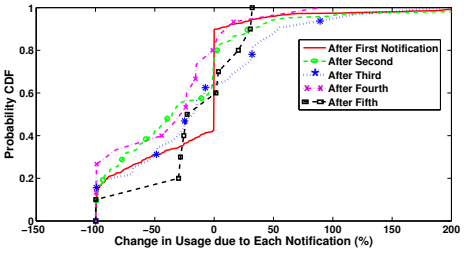
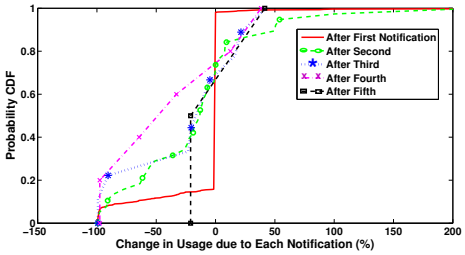


Figure 10: iPad response to static TDP.



(a) Usage changes (iPad users).



(b) Usage changes (iPhone users).

Figure 11: User response to notifications sent.

Type	Periods	First Stage		Second Stage	
		Color	Disc.	Color	Disc.
1	2, 8, 14, 20	Orange	10%	Orange	28%
2	3, 6, . . . , 24	Orange	10%	Green	30%
3	5, 11, 17, 23	Orange	10%	Orange	9%

Table 5: Period types in the color experiment.

type between the first and second stages of the experiment. To do so, we first found the average usage in each period of the day (i.e., each hour) for the first stage of the experiment. We then calculated the percent change in usage of each period in the second stage of the experiment from the average usage in the same period during the first stage. The *average change* in each type of period is then defined as the weighted average of these percent changes for each period of the given type. The weights were proportional to the usage in that period.

Results: We found that *users paid more attention to indicator color than to the numerical discount value.* When discounts increased significantly with no change in indicator color, only half of the users increased their usage relative to other periods. However, when the indicator color also changed, almost all users increased their usage in those periods relative to others. In Fig. 12, each data point represents one user's average change in each period type, with the size of the data point indicating the volume of usage in the second stage of the experiment. The reference line represents equal changes in both period types considered.

Figure 12a shows the average change in usage for each user in Type 1 periods versus Type 3 periods. For both period types, the color did not change, but the discount in Type 1 periods increased significantly. Thus, if users had reacted to the numerical prices, we would expect usage to increase in Type 1 and decrease in Type 3 periods: users' data points should lie above the reference line. Figure 12a shows that this is the case with only half of the users. Moreover, some users increased their usage dramatically in both types of periods, while most decreased their usage in both types of periods. Wilcoxon's test reveals an 82% probability that the null hypothesis is valid on the differences in usage changes in both types of periods: since the indicator color did not change, users were mostly agnostic to the numerical values of the discounts.

We next compare these results to those obtained when the indicator color and discount offered both change. Figure 12b plots the average change in usage in Type 2 versus Type 1 periods. The discounts in both periods increased by comparable amounts, but the indicator color changed from orange to green only in Type 2 periods. We see that most users' data points lie above the reference line, so that usage increased more (or decreased less) in Type 2 as compared to Type 1 periods. Wilcoxon's test yields only a 9.8% probability that the null hypothesis is valid on the differences in usage changes in both types of periods. Thus, users responded to the indicator color despite the comparable numerical discounts.

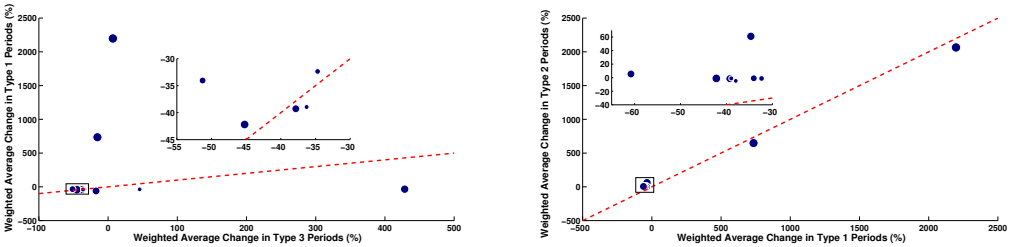
4.4.4 *Benefits of Optimized TDP*

Question: *Does peak usage decrease with time-dependent pricing? And does this decrease come at the expense of an* overall *decrease in usage?*

Method: We offered optimized time-dependent prices to all users with JB devices over a two week period in January 2012. The waiting functions used to calculate these prices were estimated from usage data for the static time-dependent prices. To measure the peak reduction, we calculated the *peak-to-average ratio* (PAR), i.e., the ratio of usage in the peak period to average per-period usage, for each day. We then compared the PARs from our TIP data to those observed with TUBE's optimized TDP.

Results: *Optimized time-dependent prices reduce the peak-to-average ratio* from TIP usage. Moreover, *overall usage significantly increased* after TDP was introduced, partially because people used more in the discounted valley periods.

Figure 13a shows the distribution of daily PARs both before and after TDP was introduced. The maximum PAR decreases by 30% with TDP, and approximately 20% of the



(a) Period types 1 and 3.

(b) Period types 1 and 2.

Figure 12: Changes in usage for the period types in Table 5.

PARs before TDP are larger than the maximum PAR with TDP. Thus, TDP significantly reduced the peak-to-average ratio, flattening demand over the day.

We next show that this decrease in PAR is not due to a net loss of traffic. Figure 13b shows the peak daily usage observed before and after TDP. Though the maximum peak usage is about the same, peak usage is generally larger with TDP. Since the PARs also decreased, usage in non-peak periods increased. In fact, we observe a 130% increase in usage from TIP to TDP. Part of this increase may be due to the time of year–we measured the TIP usage from July to September, and the TDP usage in January. TDP, however, is likely a major factor: the discounts during off-peak periods allowed users to consume more data while still spending less money and decreasing the PAR.

Finally, we examine the changes in application distribution due to the introduction of TDP. Figure 13c shows the average daily usage by application before TDP (i.e., during our TIP calibration period) and when TDP is introduced. We see that movie streaming nearly quadrupled, while usage of other applications stayed about the same. Since streaming is generally used for entertainment, the discounts may have motivated people to consume more data during low-price periods. Thus, the "valleys" of TIP usage were further filled up by an increase in demand, creating a win-win-win across end users, ISPs, and content providers.

5. DISCUSSION AND FUTURE WORK

We now discuss several limitations of TUBE's model and our trial, and then identify future extensions of our work.

Mobility: Since mobile users may switch their base stations frequently, the proposed TUBE system requires user devices to keep track of different prices across distinct base stations. However, user mobility can often be predicted from location history [22]; thus, the user device can predict its location over the next day and pull prices from the appropriate base stations. The device itself can then keep track of the user's bill, an approach scalable to many users.

Single bottleneck: Our model assumes a single bottleneck in the network. This assumption is consistent with a wireless base station in an urban area or a middle-mile bottleneck in rural areas.

Time granularity: Our initial trial uses hour-long periods, since users are more familiar with hourly prices, but we can also shrink the periods to, say, 10 minutes. Shorter time periods allow users to wait less, possibly enhancing TDP's effectiveness. If the timescale is further reduced to several seconds, the autopilot mode on user devices effectively responds to real-time congestion conditions, turning time-dependent pricing into congestion-dependent pricing.

Trial scale and additional functionalities: Different demographics likely have different price and time sensitivities for mobile data, making our trial only the first step towards understanding TDP's effectiveness. We are currently conducting larger-scale trials with U.S. and Indian ISPs that will further illuminate users' price-delay tradeoffs and directly test the effect of the autopilot mode.

Control group: While a trial control group is highly desirable, it is difficult to compare the TDP usage of one group of users with another control group of TIP users unless they are matched properly. Such matching is especially challenging in a small population of users, as in our trial. Our planned large-scale trials will address this issue.

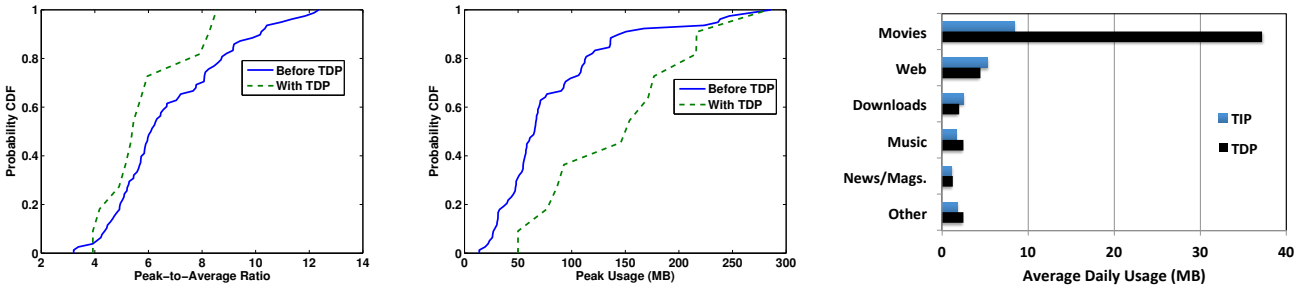
6. RELATED WORK

Internet pricing models have been debated since the 1990s. Several pricing schemes, both *static* and *dynamic*, have been suggested by networking researchers to alleviate congestion. Static pricing plans charge users according to predetermined rates without adapting to customers' usage behavior, e.g., metered, flat price, cap then metered, and two-period time of day pricing plans [17]. Other proposals include Clark's Expected Capacity Pricing [3], Cocchi's Edge Pricing [4], and Odlyzko's Paris Metro Pricing [16], with the former two admitting dynamic versions as well.

Dynamic pricing has the advantage of adapting prices to the network condition, as shown in Gupta et al.'s Priority Pricing [7], Hayer's Transport Auction [9], Kelly et al.'s Proportional Fair Pricing [11], Varian's Smart Market Pricing [13], MacKie-Mason et al.'s Responsive Pricing [14], Semret et al.'s Market Pricing [19], etc. Sen et al. [20] provides a detailed overview of these various pricing proposals and their realization in current data plans. The social and ethical dimensions of dynamic pricing have also been widely studied, and its consumer adoption and benefits in electricity and transportation networks are well documented [20]. However, there have been no documented trials of dynamic pricing for mobile data.

7. CONCLUSION

Though time-dependent pricing for mobile data has been discussed for several decades, no experimental study has been conducted to investigate a functional prototype. To this end, we developed and implemented TUBE, an architecture that takes TDP from economic theory to a system implementation. TUBE creates a feedback loop between the ISP's price computation and users' ever-changing response to these prices. To link these components, we estimate and predict users' future behavior each day with aggregate usage data. Users respond to the prices via a GUI that resides on their devices, either manually or using an autopilot mode.



(a) PARs for TIP and TDP. (b) Peak traffic for TIP and TDP. (c) App distribution with TIP and TDP.

Figure 13: Usage statistics for TIP and TDP.

To confirm TDP's feasibility, we conduct a trial with 50 iPhone/iPad 3G data plan users, acting as a resale ISP and charging them according to our TDP algorithms. Our trial results indicate that people are sensitive to time-dependent prices and indeed shift their Internet usage to off-peak periods, while increasing the monthly total of data usage. TDP flattens the temporal distribution of user demand for data, thus reducing ISP cost while allowing users to save money. This implementation and pilot trial motivates future study on time-dependent pricing for different markets and demographics.

Acknowledgements

We thank Rudiger Rill and Dana Butnariu for their iOS development efforts, Jennifer Rexford for her comments on an earlier version of this work, and many colleagues for discussions on smart mobile pricing. We also thank AT&T for providing the APN and our trial customers for their participation. Part of the work reported here was supported by NSF CNS-1117126 and AFOSR FA9550-09-1-0643. C. Joe-Wong was supported by the NDSEG fellowship.

8. REFERENCES

[1] Cheng, R. Verizon to curb highest data users, Feb. 4 2011. Wall Street Journal.

[2] Cisco Systems. Cisco visual networking index: Forecast and methodology, 2011-2016, May 30 2012. http://tinyurl.com/VNI2012.

[3] Clark, D. D. Internet cost allocation and pricing. In *Internet Economics*, L. W. McKnight and J. P. Bailey, Eds. The MIT Press, 1997, pp. 215–252.

[4] Cocchi, R., Shenker, S., Estrin, D., and Zhang, L. Pricing in computer networks: Motivation, formulation, and example. *IEEE/ACM Transactions on Networking 1* (1993), 614–627.

[5] Falaki, H., Mahajan, R., Kandula, S., Lymberopoulos, D., Govindan, R., and Estrin, D. Diversity in smartphone usage. In *Proc. of ACM MobiSys* (2010), ACM, pp. 179–194.

[6] Goldman, D. AT&T hikes rates on smartphone plans, Jan. 19 2012. CNN Money.

[7] Gupta, A., Stahl, D., and Whinston, A. Priority pricing of integrated services networks. In *Internet Economics*, L. W. McKnight and J. P. Bailey, Eds. The MIT Press, 1997, pp. 323–352.

[8] Ha, S., Sen, S., Joe-Wong, C., Im, Y., and Chiang, M. TUBE survey questions and demographics, Jan. 2012. http://scenic.princeton.edu/tube/TUBE_Survey.pdf.

[9] Hayer, J. Transportation auction: A new service concept. *M.Sc./M.B.A. Thesis, TR-93-05* (1993).

[10] Joe-Wong, C., Ha, S., and Chiang, M. Time-dependent broadband pricing: Feasibility and benefits. *Proc. of IEEE ICDCS* (June 2011).

[11] Kelly, F., Maulloo, A. K., and Tan, D. H. K. Rate control for communication networks: Shadow prices, proportional fairness, and stability. *Journal of Operational Research Society 49* (1998), 237–252.

[12] Langley, R. *Practical Statistics Simply Explained*. Dover Books Explaining Science Series. Dover Publications, 1971.

[13] MacKie-Mason, J., and Varian, H. Pricing the Internet. In *Public Access to the Internet*, B. Kahin and J. Keller, Eds. Prentice-Hall, 1995, pp. 269–314.

[14] MacKie-Mason, J. K., Murphy, L., and Murphy, J. Responsive pricing in the Internet. In *Internet Economics*, L. W. McKnight and J. P. Bailey, Eds. The MIT Press, 1997, pp. 279–303.

[15] NIST. *Engineering Statistics Handbook*. http://itl.nist.gov/div898/handbook/pmc/section4/pmc435.htm.

[16] Odlyzko, A. Paris metro pricing for the Internet. *Proc. of the 1st ACM Conf. on Electronic Commerce* (Nov. 1999).

[17] Parris, C., Keshav, S., and Ferrari, D. A framework for the study of pricing in integrated networks. Tech. rep., Tenet Group, ICSI, UC Berkeley, 1992. TR-92-016.

[18] Princeton EDGE Lab. TUBE website, 2012. http://scenic.princeton.edu/tube/.

[19] Semret, N., Liao, R. R.-F., Campbell, A. T., and Lazar, A. Pricing, provisioning and peering: Dynamic markets for differentiated internet services and implications for network interconnections. *IEEE Journal on Selected Areas in Communications 18* (2000), 2499–2513.

[20] Sen, S., Joe-Wong, C., Ha, S., and Chiang, M. Pricing data: A look at past proposals, current plans, and future trends. *arXiv* (Feb. 2012). http://arxiv.org/abs/1201.4197.

[21] Shih, J., Katz, R., and Joseph, A. Pricing experiments for a computer-telephony-service usage allocation. In *Proc. of IEEE Globecom* (2001), vol. 4, IEEE, pp. 2450–2454.

[22] Song, C., Qu, Z., Blumm, N., and Barabási, A. Limits of predictability in human mobility. *Science 327*, 5968 (2010).

[23] Taylor, J. Short-term electricity demand forecasting using double seasonal exponential smoothing. *Journal of the Operational Research Society* (2003), 799–805.

[24] Taylor, P. AT&T imposes usage caps on fixed-line broadband, March 14 2011. Financial Times.

[25] The Economist. The mother of invention: Network operators in the poor world are cutting costs and increasing access in innovative ways, Sept. 24 2009. Special Report.

[26] TUBE Project. Enabling mobile time-dependent pricing, 2012. http://www.youtu.be/lXuJw4tWH40.

[27] Varaiya, P. P., Edell, R. J., and Chand, H. INDEX project report, Aug. 1996. http://people.ischool.berkeley.edu/~hal/index-project/R98_005P.PDF.

[28] Welch, C. Verizon to kill grandfathered unlimited data plans for customers seeking upgrades, May 16 2012. The Verge.