Abstract—Charging different prices for Internet access at different times induces users to spread out their bandwidth consumption across times of the day. Potential impact on ISP revenue, congestion management, and consumer behavior can be significant, yet some fundamental questions remain: is it feasible to operate time dependent pricing and how much benefit can it bring? We develop an efficient way to compute the cost-minimizing time-dependent prices for an Internet service provider (ISP), using both a static session-level model and a dynamic session model with stochastic arrivals. A key step is choosing the representation of the optimization problem so that the resulting formulations remain computationally tractable for large-scale problems. We next show simulations illustrating the use and limitation of time-dependent pricing. These results demonstrate that optimal prices, which “reward” users for deferring their sessions, roughly correlate with demand in each period, and that changing prices based on real-time traffic estimates may significantly reduce ISP cost. The degree to which traffic is evened out over times of the day depends on the time-sensitivity of sessions, cost structure of the ISP, and amount of traffic not subject to time-dependent prices. Finally, we present our system integration and implementation, called TUBE, and proof-of-concept experimentation.

I. INTRODUCTION

A. Motivation

Internet service providers (ISPs) practicing flat rate pricing face a dilemma: unlike its cost, an ISP’s revenue does not scale with users’ ever increasing desire for more bandwidth. Usage-based pricing has long been adopted by ISPs outside the United States and, with AT&T and Verizon’s pricing plan changes, recently entered the U.S. wireless and now wireline markets (e.g., [1], [2]). Much of this is driven by the tremendous growth of network traffic, which is out-pacing the expansion of capacity and turning ISPs’ attention to pricing as the ultimate congestion management tool to regulate bandwidth demand. Yet pricing based just on monthly bandwidth usage still leaves a timescale mismatch: ISP revenue is based on monthly usage, but peak-hour congestion dominates its cost structure. Ideally, ISPs would like bandwidth consumption to be spread evenly over all the hours of the day.

Time-dependent usage pricing (TDP) charges a user based on not just “how much” bandwidth is consumed but also “when” it is consumed, as opposed to time-independent usage pricing (TIP), which only considers monthly consumption amounts. TDP has the potential to even-out time-of-the-day fluctuations in bandwidth consumption [3]. As a pricing practice that does not differentiate based on traffic type, protocol, or user class, TDP also sits lower on the radar screen of network neutrality scrutiny. In its December 2010 statement, the FCC in the U.S. encouraged “measures to match price to cost.” Time-dependent pricing also presents more choices to all consumers [4] and may mitigate potential adverse impact of TIP on the surging trend of movie streaming, cloud service, and bandwidth-intensive video advertising. In fact, the daytime (counted as part of minutes used) and evening-time (free) pricing, long practiced by wireless operators, is a simple, 2 period TDP scheme. Operators in India are already taking these plans a step farther, with time-dependent pricing for voice calls. In the United States, small ISPs in some states have begun experimenting with TDP for data traffic, although in their current implementation, users have no interface to react to the time-dependent prices, and the prices are not optimized accordingly.

Given the “time inelasticity” of bandwidth demand in different demographics and applications, it is not clear how much TDP can reduce ISPs’ costs, due to either impatient users or time-sensitive applications, such as real-time streaming or online gaming. Yet at the same time, the volume of time-elastic applications is also rising. Multimedia downloads, file sharing, social media updates, data backup, and non-critical software downloads all have various degrees of time elasticity. Can we efficiently parametrize time-elasticity and then leverage them in setting the right prices?

Even TDP’s feasibility needs examination. Research on integrating traffic measurement, optimal price determination, and user interface design is necessary for TDP to become feasible. Furthermore, it is unclear if time-dependent prices could be optimized in a computationally efficient way for near real-time control. This paper investigates how an ISP can use TDP to manage network congestion by addressing these questions. We introduce a set of algorithms to efficiently determine optimal prices, taking into account anticipated user reaction, and then present an integrated system design called TUBE (time-dependent usage-based broadband-price engineering), an end-to-end TDP system for ISPs. Figure 1 summarizes the TDP prototype as a control loop. This paper first discusses the center module of computing optimal prices and quantifies its efficacy in simulation, then explores the modules of user
profiling, measurement, user interface, and finally presents system integration and a proof-of-concept experiment.

B. Related Work

The electricity industry has explored TDP over the years, as shown in Table I’s summary of existing TDP literature. Extending these economic analyses to broadband pricing is non-trivial for several reasons:

- Our model forms part of Fig. 1’s control loop, so that ISPs can adapt prices in real time to user behavior while users react to ISPs’ prices.\(^1\)
- We model TDP as users deferring part of their Internet usage, rather than the electricity market’s model of users choosing the period in which to demand a resource.
- In prior work for the electricity industry, the bottleneck is resource generation, not transit as for ISPs. This difference requires tracking arrival and departure of application sessions as in our dynamic model.
- Previous models for broadband TDP use simplified “representative demand functions” to estimate resource demand at peak and off-peak times, while we develop detailed models directly incorporating sessions’ time-sensitivity.
- We use \(n\) (e.g. \(n = 48\) for half hour granularity) periods instead of 2; the multiple peaks and valleys in bandwidth usage over one day make 2 period TDP inadequate. Without a binary pre-classification of hours into peak and off-peak periods, the design is more challenging.

This paper’s formulation and methodology apply to both wireline and wireless pricing, and may be generalized to satellite capacity pricing and cloud service pricing. Countries outside of US may see widespread usage of TDP first. In the U.S., wireless TDP will likely take off first, given its (on average) $10/GB usage price today and the rapid growth of wireless data usage.\(^2\)

\(^1\)Many prior works on TDP for electricity do not model real-time user reaction due to the lack of a convenient graphic user interface (GUI) and the relatively low elasticity of electricity usage. In contrast, broadband TDP can readily position GUIs on Internet access devices, and the elasticity of bandwidth consumption tends to be high for a good range of applications.

\(^2\)There is also a variety of other commonly studied network economics topics, including inter-ISP pricing and its relationship to BGP, two-sided pricing where ISP charges both consumers and content providers [5], and QoS differentiation via price differentiation as in Paris Metro Pricing [6].

\[\text{Table I} \quad \text{Summary of previous papers on time-dependent pricing.}\]

<table>
<thead>
<tr>
<th>Work</th>
<th>Industry</th>
<th>Periods</th>
<th>Model Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[7]</td>
<td>Electricity</td>
<td>2</td>
<td>DF</td>
<td>SW analysis of simulation based on real data</td>
</tr>
<tr>
<td>[8]</td>
<td>Electricity</td>
<td>2</td>
<td>DFRD</td>
<td>Analysis of California pilot study</td>
</tr>
<tr>
<td>[9]</td>
<td>Electricity</td>
<td>2 or 3</td>
<td>DF</td>
<td>Various articles</td>
</tr>
<tr>
<td>[10]</td>
<td>Electricity</td>
<td>2, 24</td>
<td>DFRD</td>
<td>Pilot study proposal; previous studies reviewed</td>
</tr>
<tr>
<td>[11]</td>
<td>Electricity</td>
<td>2</td>
<td>DFRD</td>
<td>Quantitative user behavior prediction</td>
</tr>
<tr>
<td>[12]</td>
<td>Electricity</td>
<td>2</td>
<td>DF</td>
<td>Application of theoretical model to real data</td>
</tr>
<tr>
<td>[13]</td>
<td>Electricity</td>
<td>2</td>
<td>DFRD</td>
<td>Analysis of California pilot study</td>
</tr>
<tr>
<td>[14]</td>
<td>Electricity</td>
<td>(n/a)</td>
<td>Spot price pass-through</td>
<td>Cost-benefit analysis using previous trials</td>
</tr>
<tr>
<td>[15]</td>
<td>Electricity</td>
<td>2</td>
<td>DFRD</td>
<td>Analysis of Japanese results</td>
</tr>
<tr>
<td>[16]</td>
<td>Electricity</td>
<td>3</td>
<td>DFRD</td>
<td>Ontario pilot study analysis</td>
</tr>
<tr>
<td>[17]</td>
<td>Electricity</td>
<td>24</td>
<td>DF</td>
<td>Cost-benefit analysis of case studies</td>
</tr>
<tr>
<td>[18]</td>
<td>Electricity</td>
<td>2</td>
<td>DFRD</td>
<td>Anaheim pricing experiment analysis</td>
</tr>
<tr>
<td>[19]</td>
<td>ISP</td>
<td>(n)</td>
<td>Game Theoretic</td>
<td>Theoretical analysis of SW</td>
</tr>
<tr>
<td>[21]</td>
<td>General</td>
<td>(n)</td>
<td>DF with uncertainty</td>
<td>Theoretical model</td>
</tr>
<tr>
<td>[22]</td>
<td>General</td>
<td>(n/a)</td>
<td>Qualitative description</td>
<td>Argument for time-dependent pricing</td>
</tr>
</tbody>
</table>

C. Overview of Models and Summary of Results

When determining optimal prices, an ISP tries to balance the cost of demand exceeding capacity—e.g. the capital expenditure of capacity expansion—with the cost of offering reduced prices to users willing to move some of their sessions to later times. A user is a set of application sessions, each with a waiting function giving the willingness to defer that session for some amount of time and some pricing incentive for doing so. Pictorially, an ISP uses TDP to even out the “peaks” and “valleys” in bandwidth consumption over the day. The ISP’s problem is then to set its prices to balance these two types of costs, given its estimates of user behavior and willingness to defer sessions at different prices.

The ISP’s decision can equivalently be formulated in terms of rewards, as in our formulation. The ISP rewards users for deferring by the difference between TIP and optimal TDP prices. Without loss of generality, rewards are positive; their values reflect movement of the baseline usage price.

Section II develops the static model, which does not include stochastic arrival of new sessions. We prove that waiting
functions concave in rewards and a piecewise linear cost of exceeding capacity imply that price determination is a convex optimization, ensuring computational tractability.

Section III extends to dynamic models with stochastic arrivals. For a single bottleneck network, this model reduces to the static model with demand under TIP equal to the amount of traffic arriving in each period. The fixed-size version is then extended to sessions with fixed duration and online adjustment that tracks user behavior. This online algorithm is later used in the TUBE Optimizer, as in Fig. 9’s schematic.

Traditional economic models explicitly specify users’ representative demand in each period, an approximate approach not easily scalable to multiple periods. Instead, our waiting functions use only a general time-sensitivity to model users’ deferral behavior. We also consider uncertainty in user behavior: these functions give the probability that a session will defer for a given amount of time and reward. Waiting functions may be distinct for each application session or may represent an aggregate of users’ willingnesses to wait, averaged over concurrent sessions.

While the waiting functions depend on the amount of time deferred, the ISP does not need to track users’ behavior in our design—its waiting function estimation to statistically model users’ deferral behavior. Thus, all sessions in a given period are charged the same price, no matter how long they are deferred. In Section IV we give sample waiting functions, illustrating the variation in time-sensitivities and presenting a waiting function estimation algorithm. The estimation uses only aggregate, not individual, TIP and TDP usage data. The ISP only needs to record a user’s TDP usage per period in order to charge the correct amount on that user’s monthly bill.

Throughout this paper, we assume the following:

- ISPs are monopolies, facing an estimated distribution of users’ willingness to wait.

- Each session consumes a fixed amount of ISP capacity, e.g., the average over its short time-scale fluctuations.

- TDP does not cause application sessions to disappear.

Section V shows numerical simulations of the models in Sections II and III, based on empirical data from a large ISP. Section VI discusses practical aspects of implementing TDP in our system integration, called TUBE (Time-dependent, Usage-based Broadband-price Engineering). We also show a proof-of-concept experimentation with TUBE. These results confirm the feasibility of TDP in advance of a planned field trials, described in the conclusion section together with a discussion of congestion-dependent pricing in wireless networks. Details beyond the page limit here can be found in the technical report [23].

II. Static Session Model and Formulation

Different representations of the same underlying optimization problem may require different computational loads. In fact, naïve representations of several of our problem formulations would lead to non-convex, high-dimensional optimization. In contrast, our representation ensures computational tractability of ISPs’ near real-time TDP price optimization.

The ISP’s objective is to minimize the weighted sum of the cost of exceeding capacity and of offering reduced prices (i.e., rewards). The optimization variables are these rewards, which give users incentives to defer bandwidth consumption. Let $X_i$ denote period $i$ demand under TIP. The phrase “originally in period $i$” means that with TIP, this session occurs in period $i$.

Suppose that the ISP divides the day into $n$ periods, and that its network has a single bottleneck link of capacity $A$. This link is often the aggregation link out of the access network, which has limited bandwidth compared to aggregate demand and is often oversubscribed by a factor of five or more. The cost of exceeding capacity in each period $i$, capturing both customer complaints and expenses for capacity expansion, is denoted by $f(x_i - A)$, where $x_i$ is usage in period $i$. Capital expenditure cost is incurred over a large timescale; the $f$ cost function represents the fraction due to daily capacity exhaustion.

Each period $i$ runs from time $i - 1$ to $i$. A typical period lasts a half hour. Sessions begin at the start of the period, an assumption readily modified to a distribution of starting times. The time between periods $i$ and $k$ is given by $i - k$, which is the number $b \in [1, n], b \equiv i - k \pmod n$. If $k > i$, $i - k$ is the time between period $k$ on one day and period $i$ on the next.

For each session $j$ originally in period $i$, define the waiting function $w_j(p, t) : \mathbb{R}^2 \rightarrow \mathbb{R}$, which measures the user’s willingness to wait $t$ amount of time, given reward $p$. Each session $j$ has bandwidth requirement $r_j$, so $v_jw_j(p, t)$ is the amount of session $j$ deferred by time $t$ with reward $p$. To ensure that $w_j \in [0, 1]$ and that the calculated usage deferred out of a period is not greater than demand under TIP, we

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Static Model</th>
<th>Dynamic Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_i$</td>
<td>Reward for deferring to period $i$</td>
<td>Same</td>
</tr>
<tr>
<td>$x_i$</td>
<td>Usage in period $i$</td>
<td>Same</td>
</tr>
<tr>
<td>$A_i$</td>
<td>Maximum capacity in period $i$</td>
<td>n/a</td>
</tr>
<tr>
<td>$f(x)$</td>
<td>$\max (x, 0)$</td>
<td>Same</td>
</tr>
<tr>
<td>$X_i$</td>
<td>Period $i$ usage with TIP</td>
<td>Same</td>
</tr>
<tr>
<td>$w(p, t)$</td>
<td>Waiting function</td>
<td>Same</td>
</tr>
<tr>
<td>$v_j$</td>
<td>Volume of session $j$</td>
<td>n/a</td>
</tr>
<tr>
<td>$j \in i$</td>
<td>Sessions $j$ originally in period $i$</td>
<td>n/a</td>
</tr>
<tr>
<td>$i - k$</td>
<td>$i - k \pmod n$</td>
<td>Same</td>
</tr>
<tr>
<td>$\Pi_i(t)$</td>
<td>Sessions arriving in period $i$ up to time $t$</td>
<td>n/a</td>
</tr>
<tr>
<td>$M_{i,k}(t)$</td>
<td>Sessions deferring for $k$ periods from period $i$ up to time $t$</td>
<td>n/a</td>
</tr>
<tr>
<td>$N(t)$</td>
<td>Active sessions, time $t$</td>
<td>n/a</td>
</tr>
<tr>
<td>$g$</td>
<td>PDF for $w$ parameters</td>
<td>n/a</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Allocated capacity</td>
<td>n/a</td>
</tr>
<tr>
<td>$w_j(p, t)$</td>
<td>The function $\frac{p}{x_j + \mu}$</td>
<td>n/a</td>
</tr>
</tbody>
</table>

PDF: probability density function
normalize the \( w_j \), dividing by the sum over possible times deferred \( t \) of \( w_j(P, t) \). Here \( P \) is the maximum possible reward offered, or maximum marginal cost of exceeding capacity.

**Proposition 1:** The ISP’s optimization problem for time-varying rewards can be formulated as

\[
\min \sum_{i=1}^{n} p_i \left( \sum_{k=1, k\neq i}^{n} \sum_{j \in k}^{n} v_j w_j(P_i, i - k) \right) + f(x_i - A_i) \\
\text{s. t. } x_i = X_i - \sum_{j \in i}^{n} v_j \sum_{k=1, k\neq i}^{n} w_j(p_k, k - i) + \sum_{k=1, k\neq i}^{n} \sum_{j \in k}^{n} v_j w_j(P_i, i - k),
\]

\[\text{var. } p_i; i = 1, \ldots, n.\]

**Proof:** See [23]. The key step uses the waiting function normalization to track aggregate usage deferred from and into each period.

We have the following equivalence of problem formulations:

**Proposition 2:** Minimizing cost in (1-2) and maximizing profit are equivalent.

**Proof:** See [23]. The key step is writing profit with TIP as revenue minus operational cost and dividing cost into before and after exceeding capacity. Revenue with TDP is then revenue with TIP minus the cost of offering rewards.

In usage-based pricing, whether time-dependent or not, the ISP may charge a flat rate until users reach a certain cap, and after that charge a usage-based rate. Explicitly modeling this cap in TDP considerably complicates tractability of the problem, so we instead vary available capacity with time. In each period, the ISP subtracts from the network capacity \( A \) usage from those users not reaching the cap and thus not affected by TDP. This time-dependence also allows for a cushion of excess capacity against irrational users, a typical precaution for ISPs. The optimization problem then only involves sessions above the cap. Since \( A_i \), the available capacity in period \( i \), is independent of price, the model is essentially unchanged.

For efficient price determination in TDP, the optimization problem must have a scalable solution algorithm. The most useful criterion for this property is convexity: minimizing a convex function over a convex constraint set. We find mild conditions on the \( w_j(p, t) \) that make the problem (1-2) convex and accommodate different price- and time-sensitivities.

**Proposition 3:** If the \( w(p, t) \) are increasing and concave in \( p \), and \( f \) is piecewise-linear with bounded slope, the ISP’s optimization problem is convex.

**Proof:** See [23]. The key step is finding the cost function’s Hessian matrix and observing that ISPs will not offer rewards greater than the marginal benefit of reduced capacity cost.

The conditions in Prop. 3 are readily satisfied: following the principle of diminishing marginal utility, \( w_j \) should be increasing and concave in \( p \) and decrease in \( t \). Users prefer to defer for shorter times. ISP cost can also be readily represented with piecewise-linear functions of bounded slope.\(^3\)

### III. Dynamic Session Models and Formulations

#### A. Offline Model

The dynamic model has offline and online versions. The offline model uses historical demand statistics, and for a single bottleneck network is proven equivalent to the static model.

We assume that sessions arrive according to a Poisson random process, and leave as a function of the amount of bandwidth allocated to each session. This stochastic model is similar to that in the literature on congestion control (e.g., see the extensive bibliography in [24]). Each session has a fixed size, e.g. file downloads, and stays in the network until completely processed. We adopt the commonly used Poisson/exponential arrival model in the analysis, though the implementation will likely also encounter other types of arrival patterns. As with the static models, we assume a single bottleneck link. We use \( x \) to denote the number of sessions arriving on this link and \( \Lambda(x) \) to denote the bandwidth allocated to the link by the ISP.\(^4\)

We assume that users defer only once. Consider one time period \( i \), with start time \( i - 1 \) and end time \( i \), and define \( N(t) \) as the number of active sessions at time \( t \in [0, n] \). Since sessions may be partially processed, \( N(t) \) can be non-integral. We assume Poisson session arrival within the period with parameter \( \lambda_i \). Let \( \Pi_i(t) \) denote the number of sessions arriving between time \( i - 1 \) and time \( t \). Session sizes are assumed to be exponentially distributed with mean \( b \). Session arrival times are assumed to be uniformly distributed. Let \( \mu(N(t)) \) denote the bandwidth allocation in sessions per second.

**Proposition 4:** The ISP’s optimization problem in the offline dynamic model can be formulated as

\[
\min \sum_{i=1}^{n} \left( p_i \sum_{k=1, k\neq i}^{n} M_{k,i-k}(k) + f(bN(i)) \right) \\
\text{s. t. } N(t) = N(i - 1) - \sum_{k=1}^{n-1} M_{i,k}(t) + \sum_{k=1, k\neq i}^{n} M_{k,i-k}(k) + \Pi_i(t) - \int_{t}^{i \cdot t} \mu(N(s)) \, ds, \ t \in [i - 1, i] \\
M_{i,k}(t) = \int_{B} \int_{t}^{i \cdot t} \Pi_i(t) g_i(\beta) \times \frac{w_j(p_i+k;i-1+k-s)}{t - (i - 1)} \, ds \, d\beta
\]

\(^3\)Users may not always rationally follow estimated waiting times. Probabilistic waiting functions partially account for this uncertainty by assuming that users decide to defer a session with a certain probability, instead of always deferring to the period maximizing their waiting function. Alternatively, in [23], we present a “definite choice model” in which users defer to the period maximizing their waiting function. This model’s optimization problem is likely non-convex.

\(^4\)It is possible to adapt this formulation to sessions with fixed duration, e.g. streaming video [23]. These sessions stay in the network for a fixed amount of time and then leave; low bandwidth availability is reflected in sound and image quality and not session completion.
\( p_i(k), i = 1, 2, \ldots, n \) and \( k = 1, 2, \ldots, n - 1 \),
where \( M_{i,k}(t) \) denotes the number of sessions deferring from period \( i \) to period \( i + k \) between time \( i - 1 \) and time \( t \), \( g_i \) is the probability density function of the waiting functions \( w_\beta \) parametrized by \( \beta \), and \( B \) is the range of possible \( \beta \).

Proof: See [23]. It is similar to that for Prop. 1, but we must keep track of the number of sessions that have arrived and the number still in the network at time \( t \). □

For a single bottleneck network, \( \mu(N) \) is just the access link’s fixed capacity. This allows for a closed-form solution for \( N(t) \), giving the following proposition:

Proposition 5: For a single bottleneck network, the dynamic model is equivalent to the static model with uniformly distributed arrival times and leftover sessions from one period carrying over into the next period.

Proof: See [23]. The key step compares Props. 1 and 4 using a closed-form solution for \( N(t) \). The dynamic model thus retains the static model’s computational tractability. □

B. Online Model

Dynamic programming provides a way to solve the general problem in (3-5) with an online algorithm.

This system’s state variables consist of the rewards and the number of sessions remaining at the end of each period.\(^5\) The ISP chooses these rewards to minimize the function \( C_n(\vec{s}) \), where \( C_i \) is the incurred cost up to period \( i \). The reward \( p_n \) in period \( n \) is determined first, then \( p_{n-1}, \) etc.

We develop a low-complexity dynamic programming solution to the ISP’s optimization problem and provide an online algorithm for determining rewards. While sub-optimal, this algorithm is easy to implement and avoids the high dimensionality of a full dynamic programming solution.

**Online Price Determination Algorithm.**

1. Start with a set of rewards for the next \( n \) periods, determined with the static model or offline dynamic model.
2. After the first period, use the static or offline dynamic model to compute the optimal reward for the \( n \)th period after this first period, given the other \( n - 1 \) rewards.
3. After each subsequent period, compute the optimal reward for the \( n \)th period after the current one.

\(^5\)The initial state comes from using some set of initial rewards, for instance determined by optimization of the static model.

optimizing over prices, this section briefly describes an approach to estimating the \( w_j \). Our proposed algorithm requires only aggregate usage data under TIP and TDP, which can be obtained in control experiments during initial market trials before rolling out TDP. The ISP need not measure the traffic of individual users or separate traffic into different classes.

The ISP chooses a parametrized family of waiting functions and then estimates each period’s parameter distribution. From Prop. 3, these functions should be concave and increasing in \( p \) and decreasing in \( t \). One reasonable choice is \( w = C \frac{p}{(t+1)^\beta} \), where the normalization constant \( C \) depends on the cost of exceeding capacity, number of periods, and \( \beta \). The parameter \( \beta \geq 0 \) is a "patience index," with larger \( \beta \) indicating lower patience. Graphs of these \( w \) for different \( \beta \), evaluated at the same \( p \), are illustrated in Fig. 3 for a 12 period model and unit marginal cost of exceeding capacity. In practice, each application session may have a different \( \beta \), depending on factors such as the mood of the user at that time. Since the ISP sees an aggregated mix of sessions at any given time, there will be one \( \beta \) per type of application in each access network.

The ISP estimates waiting functions by observing the difference between demand under TIP and demand under TDP. Let \( T_i \) denote this difference in period \( i \). Suppose there are \( m \) types of sessions— for instance, the ten types in Section V. The variables \( \beta_{ij} \) then parametrize waiting functions for type \( j \) sessions in period \( i \). In our case, these are patience indices. The proportion of traffic taken up by each session type in period \( i \) is denoted by \( \alpha_{ij} \). The patience indices and proportions can vary in different periods; in each period, there are \( m \) of the \( \beta_{ij} \) and \( m \) of the \( \alpha_{ij} \), for a total of \( 2mn \) parameters. The amount of traffic deferred from period \( i \) to period \( k \neq i \) is then

\[
Q_{ik} = X_i \left( \sum_{j=1}^{m} \alpha_{ij} C \frac{p_k}{(k-i+1)^{\beta_{ij}}} \right),
\]

where \( C \) is the appropriate normalization constant. Each \( T_i \) is thus a linear function of the \( Q_{ik} \), yielding \( n \) linear equations in the \( \frac{n(n-1)}{2} \) variables \( Q_{ik} \). One equation is redundant, since we assume the sum of the \( T_i \) is zero (sessions never disappear). The ISP can estimate the parameters \( \alpha_{ij} \) and \( \beta_{ij} \) as follows:

**Waiting Function Estimation Algorithm.**

1. Compute the differences \( T_i \) between traffic under TIP and TDP, to obtain \( n \) linear equations for the \( Q_{ik} \).
2. Solve for \( n - 2 \) of the \( Q_{ik} \), making sure that for each period \( j \), at least one of the \( Q_{ik} \) is not solved for.
3. Plug these expressions back into the original equations for \( T_i \), so that only one equation, linear in the \( Q_{ik} \), remains.
4. This remaining equation then becomes a function of the offered rewards and the parameters \( \alpha_{ij} \) and \( \beta_{ij} \).
5. Use the TIP and TDP data for this function to estimate (e.g. with nonlinear least-squares) all the \( \alpha_{ij} \) and \( \beta_{ij} \) parameters involved in this one equation.
6. The parameter estimates give us the waiting functions.
TABLE III
ACTUAL AND ESTIMATED PARAMETER VALUES IN SIMULATION OF WAITING FUNCTION ESTIMATION.

<table>
<thead>
<tr>
<th>Period</th>
<th>Actual Values</th>
<th>Estimated Values</th>
<th>Maximum Percent Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\beta_{1i}$</td>
<td>$\beta_{2i}$</td>
<td>$\alpha_{1i}$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0.17</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2.33</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2.67</td>
<td>0.83</td>
</tr>
</tbody>
</table>

(a) Estimated waiting function.  
(b) Actual waiting function.

Fig. 2. Estimated and actual waiting functions for waiting function estimation; larger version in [23].

To illustrate this algorithm, we consider a simple example, with 2 types of sessions and 3 periods. Actual traffic proportions and patience indices given in Table III. We first solve for the $T_i$ in terms of the $Q_{ik}$. Then

$$T_i = \sum_{k=1}^{3} Q_{ik} - \sum_{k=1}^{3} Q_{ki}, \quad (7)$$

where for ease of notation we define $Q_{ii} = 0$. Taking $i = 1$ in (7), we solve for $Q_{12} = T_1 + Q_{21} + Q_{31} - Q_{13}$ and obtain

$$T_2 = Q_{23} - Q_{32} - (T_1 + Q_{31} - Q_{13}). \quad (8)$$

We now take (8) as our function of the rewards $p_i$, with parameters $\alpha_{j}$ and $\beta_{j}$. We generate data for the estimation by evaluating (8) at sets of offered rewards $p_i \in [0, 1]$. Table III shows the parameter values estimated by nonlinear least squares. The percent difference between actual and estimated waiting functions for each period remains small at under 12 percent. Estimated and actual waiting functions for period 1 are graphed in Fig. 2; other periods yield similar comparisons.

This estimation algorithm uses a baseline measure of aggregate demand under TIP for each period. To account for changes in the baseline over time, we iterate our algorithm. The ISP uses TDP data from a relatively long period of time, e.g. one week, to estimate the waiting functions. It can then take these estimated parameters as given and solve for the demand under TIP, $X_i$, in each period $i$. The $n$ equations (7) are linear equations in $X_i$, and all other variables are known. Due to noise in the data, different sets of rewards may give different $X_i$; the ISP can take an average to determine the baseline $X_i$. For instance, in our 3 period example, define $\omega_{ik}$ to be the (known) value of the waiting function in period $i$ for deferring to period $k$, at a given reward $p_k$. Then (7) becomes

$$X_1 - x_1 = X_1 (\omega_{12} + \omega_{13}) - X_2 \omega_{21} - X_3 \omega_{31} \quad (9)$$

at $i = 1$, with similar expressions for $X_2$ and $X_3$.

Since demand under TIP statistics are also used in the price determination, updated TIP estimates directly impact the optimal rewards. Estimation of waiting functions is not perfect no matter what statistical techniques are used, so the next section will also present simulations with incorrect waiting functions used by the ISP in their price optimization.

V. SIMULATION AND PERFORMANCE EVALUATION

In this section, aggregate traffic data over times of the day (the blue dotted line in Fig. 5) comes from one week of empirical traces by a large ISP. User patience data is much harder to obtain, so we sweep the waiting function distribution over a range of typical values (see Table IV) to quantify TDP’s impact. Our convex formulation of the static session model (Section II) and low-complexity dynamic programming algorithm (Section III) result in computationally-efficient solutions.

A. Static Session Model

We first set the number of periods, each period’s demand under TIP, sessions’ waiting functions, and the ISP’s cost function for exceeding capacity, and then set up the optimization problem (1-2) in a standard convex optimization solver.

We parametrize session waiting functions as in Section IV:

$$w_{\beta}(p, t) = C_{\beta} \frac{p}{(t + 1)^{\beta}}. \quad (10)$$

where $\beta = 0.5, 1, 1.5, \ldots, 5$. Table IV gives sample types of sessions with these waiting functions. For simplicity, $\omega$ have a linear price- or reward-sensitivity. Figure 3 illustrates time sensitivities for normalized waiting functions in a 12 period model. Using Table II’s notation, we define the cost function of exceeding capacity as follows:

$$f(x_i - A_i) = 3 \max [x_i - A_i, 0].$$

For illustrative purposes, we use monetary units of $0.10.

We use 48 half hour periods, starting at 12am. Table V shows the resulting demand under TIP in each period; this is typical of a system with ten users. Sessions are divided into the 10 waiting function types above; [23] gives the waiting function distributions. We set the single bottleneck link’s capacity to a constant 180 Megabytes/second (MBps).
The physical capacity of the bottleneck link may be larger, but ISPs often target the usage to be no more than 80% of the actual capacity, and we use that target as the value of \( A \).

The optimization yields an average daily cost per user of $3.26 with TDP and $4.26 with TIP (a 24% savings). Figures 4 and 5 respectively show the optimal rewards and traffic profile. Using Section II’s propositions, these rewards are both globally optimal and efficiently computed. The optimization ran in under 10 seconds on a standard laptop, so it is easily scalable to a large number of periods and many different session models when run on powerful servers by an ISP.

The ISP never offers a reward greater than $0.15, or half the maximum marginal benefit, due to the waiting functions’ linearity in \( p \). The ISP’s marginal cost of offering a reward \( p \) is \( 2pC \) for each session, where \( C \) represents the time deferred, from (10). But the maximum marginal benefit to the ISP is \( 3C \). Then since \( 2pC \leq 3C \), the maximum possible reward is \( p = 1.5 \), or in the monetary units assumed here, \( p = $0.15 \).

As intuitively expected, almost all of the periods with nonzero rewards are also under capacity with TIP. An exception is \( p_4 = $0.023 \); period 4 demand under TIP is 200 MBps. The ISP rewards users for deferring to period 4, which is close to over-capacity periods 1-3, and then rewards period 4 users for deferring to under-capacity periods 5, 6, etc. The net effect reduces period 4 demand from demand under TIP; the ISP transfers usage in two stages, though users only defer once.

We perturb period 1 demand under TIP for a 12 period model, with 220 MBps as the baseline case. Table VII shows both price change (the sum of the absolute values of baseline minus perturbed rewards), and percentage change in the cost using optimal and baseline rewards. As expected, these changes decrease for demand under TIP close to 220 MBps. The price change for increasing demand under TIP is smaller than for decreasing demand; for larger demand under TIP, the ISP would increase rewards for deferring from period 1. However, these are already high; baseline period 1 usage is over capacity. The small price changes for demands over 210 MBps yield cost changes under 0.01%.

We next suppose that demand under TIP is unchanged, but the ISP incorrectly measures users’ waiting functions. We suppose that users are less willing to defer than estimated by the ISP; the precise numbers used can be found in [23]. The rewards for deferring to and from period 1 change as in Table VIII. Rewards barely change, most likely because period 1 is immediately followed by several under-capacity periods. Thus, the patience indices of period 1 sessions do not much matter since the sessions are being deferred for a small amount of time. The technical report [23] contains more details of the results for perturbation of demand under TIP and waiting functions.

From Fig. 5, TDP for the 48 period model decreases the maximum minus minimum usage from 200 to 119 MBps. Overused periods closer to underused ones have the greatest traffic reduction; users more easily defer for shorter times. However, some periods are still over and others still under capacity. TDP cannot completely even out bandwidth usage fluctuations over a day if users are too impatient, sessions are too time-sensitive, or the cost of exceeding capacity is too low.

To measure the evening out of traffic over time, we define

<table>
<thead>
<tr>
<th>Period</th>
<th>Demand (MBps)</th>
<th>Price Change ($0.10)</th>
<th>Cost Change (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>190</td>
<td>0.2164</td>
<td>-3.75</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>0.0942</td>
<td>-1.50</td>
<td></td>
</tr>
<tr>
<td>210</td>
<td>0.0042</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>230</td>
<td>0.0041</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>240</td>
<td>0.0031</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>0.0072</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>260</td>
<td>0.0077</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

The optimized rewards are both globally optimal and efficiently computed. The optimization ran in under 10 seconds on a standard laptop, so it is easily scalable to a large number of periods and many different session models when run on powerful servers by an ISP.
TABLE VIII
OPTIMAL REWARDS ($0.10), PERIOD 1 WAITING FUNCTION PERTURBATION.

<table>
<thead>
<tr>
<th>Period</th>
<th>Original</th>
<th>Adjusted</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0.48</td>
<td>0.48</td>
</tr>
<tr>
<td>3</td>
<td>0.40</td>
<td>0.39</td>
</tr>
<tr>
<td>4</td>
<td>0.37</td>
<td>0.37</td>
</tr>
<tr>
<td>5</td>
<td>0.36</td>
<td>0.36</td>
</tr>
<tr>
<td>6-12</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 4. Optimal rewards, static session model. Rewards have an upper bound of $0.15, and larger rewards roughly correlate with higher traffic.

residue spread as the area between a given traffic profile and one with the same total usage but with usage constant across periods. Figure 5 yields a residue spread of 472.5 GB with TDP and 923.4 GB with TIP. The area between the two profiles is 450.9 GB, so 24% of traffic is redistributed over a day.

One would expect that when exceeding capacity is expensive, the ISP will offer large rewards to even out demand. Figure 6 shows residue spread with TDP versus the logarithm of $a$, where the cost of exceeding capacity is $af(x_i)$. Residue spread decreases sharply for $a \in [0,1,10]$, then levels out for $a \geq 10$. For $a \geq 10$, demand never exceeds capacity.

B. Dynamic Session Models

We finally simulate the offline dynamic model, with the same ten waiting function types. We use the waiting function distributions from the static model to describe the amount of traffic arriving in each period. We assume a single bottleneck network with constant capacity 210 MBps, so that the only differences between this and the static model are a uniform arrival time distribution and usage carrying over into subsequent periods. Marginal cost of exceeding capacity is $0.10$.

Figure 7 shows the optimal rewards, which yield an average daily cost of $0.72$ per user. We quantify the intuition that these are generally larger than in the static model (Fig. 4), where traffic did not carry over into different periods; the ISP now has more incentive to even out traffic. Indeed, rewards break the static simulation’s $0.15$ barrier. As shown in Fig. 8, traffic in nearly all periods is much reduced; deferred traffic from initially overused periods no longer carries over into subsequent periods. Residue spread decreases dramatically from 2623.1 GB with TIP to 1142.0 GB with TDP; the area between these traffic profiles is 1495.2 GB.

We now simulate the online dynamic model. Suppose that capacity is again 210 MBps, and that while running the online algorithm, the ISP finds that 200 instead of 230 MBps arrives in period 1 (under TIP; the ISP is using our waiting function estimation algorithm). Then optimal rewards for deferring from period 1 increases from $0.045$ to $0.572$. The ISP continues to determine optimal rewards for periods 2, 3, etc. These yield an average daily cost per user of $0.63$, which is 5% smaller than the cost with nominal rewards, $0.66$. Since in general all periods’ TIP arrival rates will vary, an online
VI. IMPLEMENTATION AND EXPERIMENTATION

To further evaluate feasibility and benefits of TDP, we are pursuing the following path towards deployment. First, we implemented TDP theory and algorithms in a Linux evaluation testbed, and integrated them with measurement and GUI. Second, in the local trial to be carried out in spring and summer of 2011 at Princeton, each participant’s Internet connection fee (wireline and wireless) will be paid by the TUBE project to their ISPs. The TUBE project will in turn act as an ISP to them, charging them based on TDP principles and design. Third, this will be followed by demonstration and potential adoption by those ISPs that have recently started using TDP but without optimizing the prices or enabling user reaction.

This section presents our implementation of TUBE and initial results running experiments with it.

A. Implementation and System Integration

The two main components of the TUBE prototype are the TUBE GUI (graphic user interface) and TUBE Optimizer, as in Fig. 9. This figure expands the network measurement and user interface boxes of the TDP control loop in Fig. 1.

Individual users install the TUBE GUI on their machines; the GUI shows their bandwidth usage and corresponding prices offered by the ISP. The TUBE Optimizer, run on ISP servers, measures individual usage and determines the prices being offered to the ISP users using Section III’s online algorithm.

We implemented the TUBE GUI as a loadable plugin to Ntop [25], an open-source Unix tool showing network usage. We also implemented the TUBE Optimizer on Linux systems by using IPtables to account for each user’s traffic usage.

The prices determined from the TUBE Optimizer are synced to the TUBE GUI at every period. The GUI loads a filter instructing the Pcap packet capture device to forward only the traffic it needs for accounting. It uses a Round Robin

Fig. 8. Traffic profile, dynamic session model. The traffic is greatly reduced, since deferred sessions from over-capacity periods no longer carry over into subsequent periods.

6Since Ntop runs on popular modern operating systems such as Windows, FreeBSD, MacOSX, and Linux, the TUBE GUI also runs on those platforms without modification.

Fig. 9. Overall schematic of the TUBE system architecture, expanding the network management and user interface boxes in Fig. 1.

The TUBE Optimizer consists of measurement, profiling, and price determination engines. The measurement engine keeps track of each user’s aggregate history and passes this information to the profiling engine, which estimates a patience index for the application. The recommendation engine then suggests when to use each application, considering the user’s monthly budget and the prices offered for the next 24 hours.

The TUBE Optimizer is self-contained, and the TUBE Optimizer’s profiling and price determination engines on a standard laptop. With 12 periods and 10 different types of sessions, the online price determination was completed in less than 5 seconds; with 3 periods and 2 types of sessions, the waiting function estimation was completed in less than 25 seconds. The TDP algorithm may be run in almost real-time due to the solution efficiency in Sections II and III.

B. Practical Considerations

Waiting Functions. Neither the TUBE GUI nor the TUBE Optimizer needs to keep track of when the original sessions arrive and depart, due to the statistical method in Section IV. This algorithm only requires the usage history under TIP and aggregate TDP usage data per period, which is available through measurement at the TUBE Optimizer.

Efficiency of the TUBE Optimizer. We measured the run time of the TUBE Optimizer’s profiling and price determination engines on a standard laptop. With 12 periods and 10 different types of sessions, the online price determination was completed in less than 5 seconds; with 3 periods and 2 types of sessions, the waiting function estimation was completed in less than 25 seconds. The TDP algorithm may be run in almost real-time due to the solution efficiency in Sections II and III.

Security and Privacy. The TUBE communication engine sends the prices determined from TUBE Optimizer to TUBE GUI through a secure SSL/TLS connection. For security and scalability of the systems, the TUBE GUI pulls the price information only once in each period. The billing data of an ISP should be protected from unauthorized access. The user profile data resides in the TUBE GUI and is not exchanged. The TUBE GUI is self-contained, and the TUBE Optimizer keeps the usage and price (reward).
C. Experimental Results

As a proof-of-concept emulation before the planned real-user trial, we test the TUBE implementation with two types of users. Users in group 1 are less patient than those in group 2. We include background traffic fluctuation at the bottleneck link too. The topology is shown in Fig. 10.

Figure VI-C shows a typical TIP traffic pattern over one hour, drawn from our TUBE testbed. Traffic is high at the beginning of the hour for both users, but lower at the end. In Fig. VI-C, user 1 never defers due to high patience indices compared to the amount of reward offered. User 2 defers; total traffic volume moved by TDP is 143.2 MB for web traffic, 707.8 MB for ftp, and 8460.7 MB for streaming video. Thus, user 2’s patience index for video is lower, corresponding to watching videos for pleasure. The amount of traffic evened out compares well with Section V’s simulations.

The bandwidth of the bottleneck is set to 10 MBps and the buffer size is set to 120 packets. The background traffic flows are generated based on the parameters used by the recent study [27] and the per-flow delays are assigned to these flows based on the empirical distribution from an Internet measurement study [28].

VII. EXTENSIONS AND CONCLUDING REMARKS

There is a tradeoff between price-sensitivity and delay-tolerance. TDP quantifies and exploits this tradeoff via adaptive pricing to create a win-win: ISPs can better manage their revenue-cost balance, consumers are presented with choices, and industries relying on heavy Internet traffic can blunt the blow from TIP usage based charging.

This paper develops the models, formulations, algorithms, system design, and prototype of a TDP system. We construct a computationally tractable price optimization framework for time-dependent, cost-minimizing pricing for ISPs. Using the proposed static and dynamic models and sweeping over a range of waiting function mixes, the ISP can solve an offline, convex optimization problem for optimal time-dependent prices. We then develop an online model that uses real-time user behavior to adjust the prices, and also present an algorithm to estimate waiting function parameters and underlying TIP usage. Using empirical time-of-the-day patterns in bandwidth consumption, our numerical simulations illustrate how much TDP with optimized prices can help even out the traffic, reduce residue spread, and reduce ISP cost.

Our TUBE implementation describes the architecture for a practical deployment.

We plan to deploy TUBE in two steps over the next few years: first, with a user trial at Princeton, and then a larger trial with partner ISPs in the United States and India. During the Princeton trial, we will act as a resale ISP: we pay participants’ monthly bill to AT&T, $10/GB in the TIP approach of usage based pricing, and the participants pay us according to the TDP system we have developed above. As shown in Fig. 13, participants will install the GUI on their handheld devices such
as iPhones, iPads, and Android phones. AT&T tunnels their traffic from their 3G core network into the servers at Princeton EDGE Lab. After this initial trial, we will partner with large American and Indian ISPs to test TUBE on a broader customer base.

Time-dependent pricing can be further extended to congestion-dependent pricing when TDP’s timescale is very short. Periods may be several seconds in wireless Internet access, where channel conditions or mobility may rapidly change congestion conditions. In such cases (and for general timescales), TDP can be put on “auto-pilot” mode, where a user need not be bothered in real time once she preconfigure her usage requirements and expectations, e.g. the maximum monthly bill, which applications should never be deferred, etc.

Pushing the auto-pilot, fast-timescale, wireless TDP approach further, ISPs can offer ultra-affordable Internet access plans, where users wait for time slots in which congestion conditions and prices are sufficiently low. Even during busy hours and over heavily used spectra, every now and then there are periods of time with little usage. We call these flashy whitespaces. Very low prices can be announced during these lightly used seconds and minutes. Those users waiting only for flashy whitespaces form a new class of “scavenger” users, trading off delay tolerance for affordable data plans.

In addition many rural users do not have wireless coverage due to several cost issues, including the cost of the middle mile backhauling traffic from the cell towers. TDP parameters can be tuned to reduce the amount of middle mile capacity needed to reach rural areas by even-ing out traffic throughout the day. Time-dependent, and congestion-dependent pricing have the potential to help bridge both urban and rural digital divides.

ACKNOWLEDGMENT

We are grateful to discussion and collaboration with the U.S. National Exchange Carrier Association, Qualcomm, and AT&T. This work is in part supported by NSF CNS-0905086.

REFERENCES