Demo: Pricing by Timing of Mobile Data

Sangtae Ha*, Soumya Sen*, Carlee Joe-Wong†, Justin Mifkovich*,
Rüdiger Rill*, Youngbin Im*, Dana Butnariu‡, Jasika Bawa*, Mung Chiang*
*Department of Electrical Engineering, Princeton University, Princeton, NJ 08544
‡Department of Computer Science, Princeton University, Princeton, NJ 08544
†Program in Applied & Computational Mathematics, Princeton University, Princeton, NJ 08544
Corresponding Author: chiangm@princeton.edu

***Abstract*—ISPs around the world are currently exploring new pricing strategies, such as usage-based pricing in the U.S., on their wired and wireless networks. A next step in this evolution of pricing practices is Time Dependent Pricing (TDP) [1], [2], which creates a win-win opportunity for ISPs and consumers: ISPs can reduce cost and monetize bandwidth, while consumers save money by choosing the time of usage. TDP uses a feedback control loop between an ISP and its users to account for users' responses to offered prices in optimizing the future prices. Implementing a functional and complete system for TDP of mobile data requires several architecture and design choices, for both the ISP-side architecture and the user interfaces (UIs). We have completed the first-of-its-kind implementation of such a system, and are conducting a user trial at Princeton. Our implementation is currently being tested by our partner ISPs [3]. This demo will introduce the audience to our system's server- and client-side features. In particular, the audience will view network congestion conditions and respond to the offered TDP prices by using our client-side UIs for Android, iOS, and Windows devices.**

I. Introduction

Wireless ISPs continue to witness an exponential surge in capacity demand from new devices, capacity-hungry applications, 4G LTE services, and cloud-based services. In an attempt to match their price to cost, major U.S. ISPs, including AT&T and Verizon Wireless, have already moved from flat rate pricing towards usage-based pricing. But usage-based pricing still falls short of solving ISPs' problem: usage fees impose costs on all users regardless of the network congestion conditions, ignoring the problem of heavy users tending to congest the network at peak periods of the day. ISPs' costs are mainly driven by these demand concentrations at certain times of the day, forcing them to over-provision their networks accordingly. An efficient solution to this problem is Time Dependent Pricing (TDP) which uses price discounts to induce users to shift their less time-sensitive demand to periods of lower congestion, thereby flattening out the demand profile and enabling temporal multiplexing.

Intuitively appealing, there are several technological, economic, and social challenges in realizing a system for TDP: How should ISPs compute optimal prices that their users will accept? How should system functionalities like price calculation and feedback be distributed between the ISP backend and the end-user device? How can we design easy-to-use user interfaces (UIs) for various device platforms?

We have recently developed an end-to-end system that addresses these issues in offering TDP for mobile data. Figure 1 shows the essential modules of our TDP system and the feedback control loop. We use economic models to profile usage behavior based on traffic measurements, then use these profiles to calculate the future prices offered. Users' reactions to these prices are recorded via the traffic measurement, and are then used to update the prices. The theory and implementation details are discussed in [1], [2]. Our system has been in trial for the last 7 months at Princeton University, and we will be running trial deployments with several ISP partners, both within and outside of the U.S. In the Princeton trial, we act as a resale ISP to the trial participants, paying their regular Internet bills and charging them according to our TDP prices.

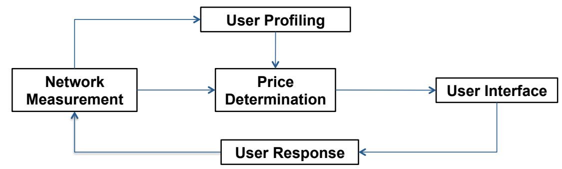

Fig. 1. Schematic of our TDP pricing system.

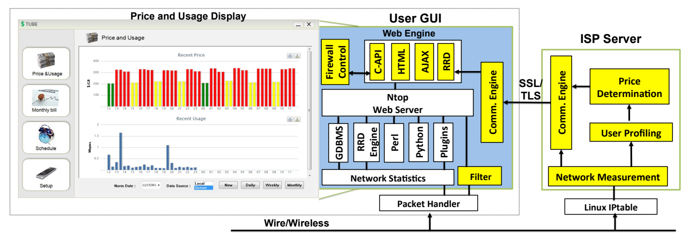

Fig. 2. TDP server- and client-side architecture.

The goal of this demo is to illustrate the inner workings of our TDP system, including how the price computation is done, how it evolves in response to usage behavior, and how users view and respond to the prices offered. In particular, we will showcase our UIs for smartphones, tablets, and notepads on the iOS, Android, and Windows platforms. The user can view her pricing and usage history, see the top bandwidth-consuming applications, and run our "autopilot" mode to automatically schedule applications so as to keep users below their target monthly budgets. On the server side, users can view the temporal evolution of congestion conditions and the TDP prices computed for future periods.

II. System Overview

The TDP system consists of server- and user-side system components as shown in Figure 2. The server, located in our

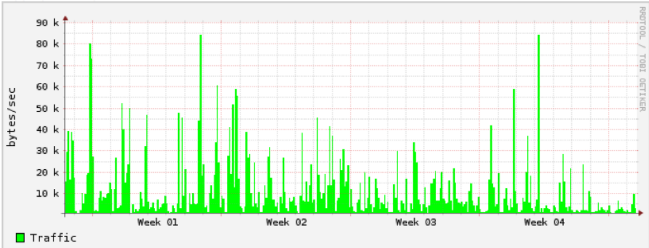

Fig. 3. Changing temporal congestion pattern from the Princeton trial.

Princeton Edge Lab, measures the aggregate traffic from all the users and estimates users' delay tolerances for various traffic classes by comparing user responses to TDP prices with that of pre-TDP periods. Different types of applications (e.g. software downloads versus email) have different tolerances for delay; we quantitatively estimate these delay tolerances, updating them as users continue to respond to the offered TDP prices. Our models are then used to compute the prices offered for the future time periods. These prices minimize the sum of two costs: that of exceeding network capacity in different time periods, and that of offering monetary discounts to users.

The client side consists of a UI installed on users' handheld devices, which serves two main purposes: self-education and automatic application scheduling. The education components allow users to view future prices, as well as the past prices and corresponding device usage. Users can also view the amount of bandwidth used by different apps, and in particular can see which apps use the most. Finally, users can set their monthly budget and turn on the "autopilot" mode, which automatically schedules applications. Users can specify delay tolerances for different applications, so that the autopilot can schedule the most delay-tolerant apps first.

III. Demo Overview

This demo will showcase the complete implementation of our TDP system on both the ISP and user sides.

For the purpose of the demo, we will act as a resale ISP and use our Princeton server as the backend that will compute and offer the TDP prices to the client devices. We show the link congestion measures via a web portal, which displays the temporal demand pattern and the amount of traffic, as in Fig. 3. The traffic shown here is the actual usage from 25 Princeton trial participants' iPhones and iPads. Users can view the traffic pattern on granularities ranging from 10 minutes to 1 day time intervals; the network usage history goes back 2 months.

The second part of the demo will showcase the user interface that we have developed on the iOS, Android, and Windows platforms. We will distribute iPads, iPhones, and Galaxy tablets to visitors, allowing real-time interaction with our GUI on these mobile devices.

Figure 4 shows screenshots of our Android UIs. On the home screen in Fig. 4a, users can see a graph of the prices offered over the next twelve hours. The prices are also displayed as a scrollable list at the top of the screen. To learn about their spending habits, users can view the history of their usage and prices on a daily, weekly, or monthly basis; Fig. 4b shows the

(a) Price display. (b) Monthly history. (c) Top five apps.

Fig. 4. Screenshots of the Android app. Users can view the future prices offered, their price and usage history, and the top 5 apps by bandwidth usage.

(a) Delay indices. (b) Monthly budget. (c) Autopilot scheduling

Fig. 5. Screenshots of the autopilot mode of Android app. Users can specify each app's delay tolerance, track monthly spending and automated scheduling.

screen for viewing monthly usage at a daily granularity. The amount of traffic used by each app is shown in Fig. 4c.

Figure 5 shows screenshots of the "autopilot" component of our UIs. Users can adjust their delay tolerances of different apps by sliding the delay tolerance bars in Fig. 5a. To aid in adjusting the delay tolerances, the apps are ranked by decreasing usage volume. Since autopilot only considers the top applications, this display makes it easy for the user to improve the scheduling recommendations of the autopilot algorithms. Users can also add to their monthly budget using the screen in Fig. 5b. The bottom graph shows the budget consumption over the month, as well as the amount of money remaining on the fuel gauge. Clicking the "Add Money" button brings up a display that allows users to add to their budget; this will result in autopilot's scheduling fewer applications. Finally, Fig. 5c shows the scheduling of different applications.

Acknowledgments

We would like to thank our ISP partners. This work has been in part supported by NSF Grant CNS1117126. C. J.-W. acknowledges support from the NDSEG fellowship.

References

[1] C. Joe-Wong, S. Ha, and M. Chiang, "Time-Dependent Broadband Pricing: Feasibility and Benefits," *Proc. of ICDCS*, June 2011.

[2] S. Ha, C. Joe-Wong, S. Sen, and M. Chiang, "Pricing by timing: innovating broadband data plans," *SPIE OPTO Broadband Access Communication Technologies VI Conference*, January 2012, paper 8282-12.

[3] "TUBE Website," http://scenic.princeton.edu/tube/.