J Comb Optim
DOI 10.1007/s10878-007-9111-3

Traffic regulation with single- and dual-homed ISPs under a percentile-based pricing policy

Jianping Wang · Jing Chen · Mei Yang · S.Q. Zheng

Abstract We investigate how a customer (an enterprise or a large organization), when facing a percentile-based pricing policy, can optimally balance the Internet access cost and the traffic buffering delay penalty by traffic regulation. The problem is referred to as the Optimal Traffic Regulation (OTR) problem. Solutions to various cases of the OTR problem are provided. For a customer with a single-homed ISP, we present optimal solutions to the OTR problem based on dynamic programming for the offline case with a known traffic demand pattern. A real-time traffic scheduling algorithm is proposed to deal with the online case where the traffic demands are different from a given demand pattern. We further extend the dynamic programming model to the case of dual-homed ISPs. Experimental results on the data from an Internet trace confirm the effectiveness of our solutions.

Keywords Internet service provider · Network management · Percentile-based pricing · Multi-homing · Optimization

Part of the results has been presented in the First International Conference on Scalable Information Systems.

J. Wang (✉)
Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong
e-mail: jianwang@cityu.edu.hk

J. Chen · S.Q. Zheng
Department of Computer Science, The University of Texas at Dallas, Richardson, TX, USA

J. Chen
e-mail: jchen602@yahoo.com

S.Q. Zheng
e-mail: sizheng@utdallas.edu

M. Yang
Department of Electrical and Computer Engineering, University of Nevada, Las Vegas, NV 89074, USA
e-mail: meiyang@egr.unlv.edu

1 Introduction

Internet service providers (ISPs) measure the usage of IP network for billing in different ways. Residential ISPs typically offer a flat rate for unlimited usage while wireless ISPs typically charge customers based on traffic volume. In the literature, there has been a lot of work on the Internet pricing; e.g. (Antoniadis et al. 2004; Cao et al. 2002; Courcoubetis and Weber 2003; Fulp and Reeves 2004; Keon and Anandalingam 2003; Li et al. 2004; Odlyzko 2004; Ros and Tuffin 2004; Wang and Schulzrinne 2000; Yaiche et al. 2000). Most of past work concentrates on the design of different pricing schemes, especially dynamic pricing schemes that accommodate various QoS services.

Recently many ISPs start to charge customers based on a statistical method called "95th percentile" (Cerruti and Wright 2002; CFDynamics 2007; Net1plus.com 1995; Service Level Corporation 2002; The Internet NG Project 2002). In general, under an α-th percentile pricing policy, an ISP records the traffic volume generated by a customer for each small time period, typically every five minutes. At the end of charging horizon, the traffic volumes in all periods are lined up from the highest to the lowest, and the α-th percentile of all five-minute traffic volumes is used as the charging volume. For example, suppose that an ISP uses a 95th percentile pricing policy, and the charging horizon for a customer contains 100 five-minute time periods. The ISP will ignore the first 5 highest traffic volumes, and use the 6th traffic volume to bill the customer. It is noticeable that the percentile-based pricing policy is also used between ISPs.

One of the major reasons that ISPs adopt such a pricing policy is that the customers are actually charged based on an approximation of their peak time traffic. This is important to ISPs because peak time traffic makes a big contribution to their capacity planning cost. Without adequate capacity planning, network performance becomes unpredictable.

From the customers' perspective, a percentile-based pricing policy is acceptable and attractive because it can tolerate their untypical traffic bursts without financial penalty. For example, under the 95th percentile billing policy, a customer can receive approximately the highest 36 hours of bandwidth usage free for each month. For a customer with a high traffic volume, such as a large corporation or a university, usually the daily traffic demand has a fixed pattern. Given its traffic pattern, the customer can actually set up a hedging point in terms of the traffic volume for each five-minute time period, and only allow $(100-\alpha)\%$ of the total time periods to have a real traffic volume greater than the hedging point. In this way, the hedging point will be used as the charging traffic volume, which is controllable to the customer. However, there may be more than $(100-\alpha)\%$ time periods with the traffic demand greater than the hedging point, and thus some traffic demand has to be delayed to the following time periods. Therefore, a customer can control its traffic to achieve tradeoffs between a lower hedging point with a lower cost and more delayed traffic demand, and a higher hedging point with a higher cost and less delayed traffic demand, without violating its bandwidth capacity constraint.

Such traffic regulation for customers with multiple Internet connections is also important because, with explosive growth of the Internet, having multiple connections

has become an essential part of businesses. Many enterprises have deployed multi-homed load balancing gateways between the ISPs and enterprise networks to increase productivity, reduce the risk of potential catastrophe, and provide fault-tolerance in case of failure of any connection.

In this paper, we study the problem of how a customer (referring to an enterprise or a large organization), when facing the percentile-based pricing policy, can optimally balance its Internet access cost and traffic buffering delay penalty with both single-homed and multi-homed ISPs by regulating its traffic. In the following, we refer to this problem as the Optimal Traffic Regulation (OTR) problem. We address both offline and online cases of the OTR problem for a customer with single-homed or multi-homed ISPs. For a customer with a single-homed ISP, we present optimal solutions based on dynamic programming model for the offline case with a known traffic demand pattern. Note that such an assumption is reasonable for those large organizations which have traffic demand patterns relatively stable over time. A real-time traffic scheduling algorithm is proposed to deal with the online case where traffic demands are different from the given demand pattern. We further extend our solutions to the case of dual-homed ISPs where the traffic is distributed to two ISPs, each employing a percentile-based pricing policy.

The rest of the paper is organized as follows. Section 2 overviews related work in traffic regulation and multi-homing. In Sect. 3, we formally define the OTR problem. In Sect. 4, we discuss the solutions to two offline cases. In Sect. 5, we show how to generalize the offline results to handle the online case. In Sect. 6, we extend the results to the case of dual-homed ISPs. In Sect. 7, we propose a problem decomposition solution to handle large-scale problems with a large charging horizon. Experimental results are reported in Sect. 8. We conclude the paper in Sect. 9.

2 Related work

Traffic regulation, which allows network administrators in an enterprise network to define how much bandwidth that a user or an enterprise network can use, is supported in most gateways available in the market, e.g., Cisco routers supporting QoS. Traffic regulation is usually supported by two different approaches: rate-limiting and traffic shaping (Cisco 2006). The rate-limiting approach drops traffic based upon rules while the traffic shaping approach generally buffers the excess traffic while waiting for the next open interval to transmit data. Both approaches identify when traffic exceeds the thresholds set by the network administrators. Traffic shaping is usually performed at the edge of the network (on customer premises) to make sure the customer is utilizing the bandwidth for business needs. Our proposed work falls into the traffic shaping category.

In the literature, traffic regulation has been studied from different perspectives, for example, under the filtering theory (Chang 1998; Chang et al. 2002), in multimedia applications (Salehi et al. 1998), and for QoS control in WDM networks (Ma and Hamdi 2000). To our best knowledge, no work has been done on traffic regulation for the purpose of Internet access cost reduction.

The research work on the impact of multi-homing under percentile-based pricing policy can be found in (Goldenberg et al. 2004; Shakkottai and Srikant 2005; Wang

et al. 2005). In (Shakkottai and Srikant 2005), the paper examines how the transit and customer prices are set in a network consisting of multiple ISPs. In (Goldenberg et al. 2004), Goldenberg *et al.* investigate how to distribute the traffic with minimum cost among multiple ISPs. In (Wang et al. 2005), the problem of how to select the ISPs to subscribe from a set of feasible ISPs is studied. It is worth pointing out that, different from the aforementioned work on multi-homing, we study the optimal traffic regulation where the customer can reduce the Internet access cost by delaying the traffic in some periods.

3 Problem description for the case of single-homed ISP

In this section, we formulate the optimal traffic regulation problem for the case of single-homed ISP. Assume that the charging horizon contains T periods, where the traffic demand for each time period t during the charging horizon is known as d_t. Let the real traffic volume sent to the ISP be x_t for period t. Under the α-th percentile pricing policy, the ISP will bill the customer based on the $\lceil (1-\alpha\%)T \rceil$-th largest x_t, which is denoted as $\bar{x}$. In other words, there will be no more than $\lceil (1-\alpha\%)T \rceil$ periods which have the real traffic volume greater than $\bar{x}$. We refer to such periods as *peak periods* and use $N = \lceil (1-\alpha\%)T \rceil$ to denote the maximum number of peak periods. Note that N is solely determined by α for any fixed charging horizon.

If, for a period t, $d_t > x_t$, then some traffic demands will be delayed to the following period, causing a deteriorative service quality. We assume that there is a penalty for each delay occurrence. For a customer who would like to spread out the traffic in favor of minimizing the cost, it is important to minimize the total delay penalty over the entire charging horizon.

The following notations are needed for defining the OTR problem.

D_{t_1,t_2}: the total traffic demand from period t_1 to t_2, $D_{t_1,t_2} = \sum_{\tau=t_1}^{t_2} d_\tau$.

y_t: the amount of traffic to be delayed from period t to period $t+1$ where $y_0 = y_T = 0$.

$f(y_t)$: the delay penalty function which is nondecreasing for $y_t > 0$ and $f(0) = 0$.

U_t: a binary variable indicating whether period t is a peak period, and $U_t = 1$ if and only if $x_t > \bar{x}$. Under the percentile-based pricing policy, we have $\sum_{t=1}^{T} U_t \le N$ for a given $\bar{x}$ and N.

$C(\bar{x})$: the cost charged by the ISP with $\bar{x}$ as the charging volume.

$\bar{B}$: the maximum capacity that the ISP can provide for a single period.

Two objectives are considered in the OTR problem: the cost charged by the ISP, and the service quality measured by the delay penalty. Considering both criteria, the OTR problem for the case of single-homed ISP (referred to as P problem in the following text) is defined as to

$$\min Z = \eta C(\bar{x}) + (1-\eta) \sum_{t=1}^{T} f(y_t)$$

subject to

$$y_t = y_{t-1} + d_t - x_t, \quad \text{for } t = 1, \ldots, T, \tag{1}$$

$$U_t = \begin{cases} 1 & \text{if } x_t > \bar{x}, \\ 0 & x_t \le \bar{x}, \end{cases} \tag{2}$$

$$\sum_{t=1}^{T} U_t \le N, \tag{3}$$

$$0 \le x_t \le \bar{B}, y_t \ge 0, \quad \text{for } t = 1, \ldots, T. \tag{4}$$

In the objective function, η, $0 \le \eta \le 1$, is a parameter showing the relative importance of the two criteria. For the constraints, (1) defines y_t which satisfies flow conservation, (2) and (3) together define the percentile-based pricing policy, and (4) enforces that the maximum capacity cannot be exceeded for each period.

In the formulation, we use a function $f(y_t)$ to represent the delay penalty of y_t units of traffic for one period. Note that this is only a high-level estimation of delay penalty. It does not necessarily mean that there are y_t units of traffic to be exactly delayed by one period. For example, suppose that at the end of a five-minute time period [13:00–13:05], we have $y_t = 200$ MB. It means at 13:05 there are 200 MB traffic being delayed and buffered. Assuming that an FIFO scheduling rule is used at the user's traffic regulation gateway, then we know that the delayed 200 MB traffic will be processed immediately at the beginning of the next time period [13:05–13:10]. Therefore, the traffic is not really delayed by five minutes. It is worthy of mentioning that even if y_t is 0, there may also be traffic being delayed within [13:00–13:05] due to the maximum available bandwidth provided by the ISP. In summary, y_t is actually a snapshot of delayed traffic at the end of every period t, and can be regarded as a good approximation of the real delay situation.

4 Offline traffic regulation for the case of single-homed ISP

4.1 Without capacity constraint

We start with a simple case where the capacity $\bar{B}$ is large enough so that it has no impact to the problem. We first study a variant of the problem P which objective is to minimize the total delay penalty subject to a maximum cost constraint. Specifically, we are given a maximum budget to be paid to the ISP, which is corresponding to a given $\bar{x}$. Then the problem is equivalent to determining x_t for each period t such that the total delay penalty $\sum_{t=1}^{T} f(y_t)$ is minimized for a given $\bar{x}$. We denote this problem as $P1$, and will show that it can be solved by a dynamic programming algorithm. Then we will discuss how to solve problem P based on the solution to problem $P1$.

We have the following theorem that describes the property of an optimal solution to problem $P1$.

Theorem 1 *For problem $P1$ without capacity constraint, there is an optimal solution where a peak period t has no traffic delayed to the next period, i.e., $x_t > \bar{x}$ implies $y_t = 0$.*

Proof If $x_t > \bar{x}$ and $y_t > 0$, we can increase x_t so as to make $y_t = 0$ since there is no capacity constraint. Such a change will reduce the traffic delay without causing any more peak periods. □

The property given by Theorem 1 results in the following dynamic programming algorithm.

The idea of our dynamic programming is to divide the entire charging horizon into a series of consecutive sub-horizons, each containing several consecutive periods among which at most one is a peak period. The problem of minimizing the total delay penalty for a given $\bar{x}$ can be solved by finding the right combination of the sub-horizons such that the total delay penalty over the entire charging horizon is minimized. Theorem 1 implies that we only need to consider those sub-horizons in which the peak period, if there is indeed one, will appear at the end of the sub-horizons.

Given such a sub-horizon from period t_1 to period t_2, it is easy to calculate the delay penalty over the sub-horizon since all traffic has to be delayed into the last period in the sub-horizon. Specifically, for $1 \le t_1 \le t_2 \le T$, let $g(t_1, t_2)$ be the minimum total delay penalty for a sub-horizon from period t_1 to t_2 where (1) there is no delayed traffic carried over to period t_1, and (2) only one potential peak period is allowed to occur at period t_2. We can use $z(t_1, t_2)$ to denote the traffic demand that has to be delayed from t_1 to t_2 with only t_2 possibly being a peak period. Then we have $z(t_1, t_1) = 0$, and for $t_2 = t_1 + 1, \ldots, T$,

$$z(t_1, t_2) = \max\{z(t_1, t_2 - 1) + d_{t_2-1} - \bar{x}, 0\}.$$

Thus we can calculate $g(t_1, t_2)$ as $g(t_1, t_2) = g(t_1, t_2 - 1) + f(z(t_1, t_2))$ where $g(t_1, t_1) = 0$.

Furthermore, we can check whether t_2 is a peak period for the above $g(t_1, t_2)$, which is denoted by a binary function $\delta(t_1, t_2)$ as

$$\delta(t_1, t_2) = \begin{cases} 1 & \text{if } z(t_1, t_2) + d_{t_2} - \bar{x} > 0, \\ 0 & \text{if } z(t_1, t_2) + d_{t_2} - \bar{x} \le 0. \end{cases} \tag{5}$$

Let $G(t, n)$ be the minimum total delay penalty from period t to period T where (1) there is no delayed traffic carried over to period t, and (2) there are no more than n peak periods from period t to period T. Consider the sub-horizon starting at period t. We can use the following dynamic programming recursion to enumerate all possible cases of the end period τ of the sub-horizon, and find the best one, i.e.,

$$G(t, n) = \min_{\tau}\{g(t, \tau) + G(\tau + 1, n - \delta(t, \tau)) | \tau = t, t + 1, \ldots, T\}. \tag{6}$$

We have the initial condition as $G(T + 1, n) = 0$ for any $n \ge 0$, and the boundary condition as $G(t, n) = +\infty$ for any $n < 0$. The optimal solution to problem $P1$ can be obtained after calculating $G(1, N)$.

The computational complexity of the algorithm is analyzed as follows. Note that although calculating a single $g(t_1, t_2)$ or $\delta(t_1, t_2)$ needs $O(T)$ time, calculating all $g(t_1, t_2)$ and $\delta(t_1, t_2)$ can be done in an incremental way. In other words, each $g(t_1, t_2)$

and $\delta(t_1, t_2)$ can be obtained in constant time from $z(t_1, t_2)$ and $g(t_1, t_2 - 1)$. So calculating all $g(t_1, t_2)$ and $\delta(t_1, t_2)$ takes $O(T^2)$ time. With all $g(t_1, t_2)$ and $\delta(t_1, t_2)$ known, each $G(t, n)$ in (6) can be done in $O(T)$ time. Because the number of $G(t, n)$ is bounded by $O(TN)$, the overall time complexity of the algorithm is in $O(T^2N)$.

To help understand the dynamic program, we give the implementation details and pseudo code for the algorithm in Appendix 1.

Now we are ready to show how to solve the original problem P. This can be done by enumerating different $\bar{x}$'s where for each $\bar{x}$, a minimum delay penalty can be obtained by solving an instance of $P1$. In many cases, the charging function $C(\bar{x})$ is a stepwise function that changes values only at a set of discrete $\bar{x}$ values such as $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{\max}$ $(=\bar{B})$. So the searching effort is limited.

4.2 With capacity constraint

Now we consider the case that the ISP capacity $\bar{B}$ has to be taken into account. As in the case without capacity constraint, we start with the problem, denoted as problem $P2$, with the objective of minimizing the delay penalty subject to a given budget, or a given $\bar{x}$, under the constraint of $\bar{B}$. Then the original problem P with capacity constraint can then be solved by enumerating all possible $\bar{x}$'s.

Before presenting the solution procedure, we first observe that in any feasible solution to the problem, the entire charging horizon can be divided into a series of consecutive sub-horizons in such a way that for each sub-horizon $[t_1, t_2]$, (1) there is no traffic delay carried over into and out of the sub-horizon, i.e., $y_{t_1-1} = y_{t_2} = 0$; and (2) there is a non-zero traffic delay between two consecutive periods within a sub-horizon, i.e., $y_\tau > 0$ for $\tau = t_1, \ldots, t_2 - 1$. Note that our definition of sub-horizon includes two extreme cases: the case that a sub-horizon may only contain a single period $[t_1, t_1]$ with $y_{t_1-1} = y_{t_1} = 0$, and the case that the entire charging horizon may be only a single sub-horizon where there is traffic delay for each period.

We notice that Theorem 1 is not applicable for problem $P2$. However, we have a modified version of the theorem as follows. We call a period t a *fractional period* if $\bar{x} < x_t < \bar{B}$.

Theorem 2 *There exists an optimal solution to problem $P2$ in which $y_t > 0$ implies that either $x_t = \bar{x}$ or $x_t = \bar{B}$.*

The proof is similar to Theorem 1, and thus omitted. Theorem 2 implies that in an optimal solution to problem $P2$, for any $\tau = t_1, t_1 + 1, \ldots, t_2 - 1$ in a sub-horizon $[t_1, t_2]$, we have either $x_\tau = \bar{x}$ or $x_\tau = \bar{B}$. If this is not true, for example, $x_\tau < \bar{x}$ or $\bar{x} < x_\tau < \bar{B}$, then according to Theorem 2, y_τ must be zero, which is inconsistent with the definition of a sub-horizon. In other words, only the last period in a sub-horizon may be a fractional period. This conclusion will help us to efficiently calculate the delay penalty for the sub-horizon.

For problem $P2$, since a sub-horizon may have multiple peak periods anywhere within the sub-horizon, which makes the solution more complex than that to problem $P1$. In the following, we propose a two-layer algorithm, where the outer layer is a dynamic program similar to (6) for finding the best combination of sub-horizons, and

the inner layer is another dynamic program for calculating the optimal solution for each sub-horizon. Theorem 2 helps us to cope with the inner layer dynamic program.

We start with the introduction of the outer layer. Let $\bar{g}(t_1, t_2, m)$ be the minimum delay penalty for a sub-horizon $[t_1, t_2]$ when there are no more than m peak periods being used. Let $\bar{G}(t, n)$ be the minimum total delay penalty from period t to T where (1) there is no delayed traffic carried over to period t, and (2) the number of peak periods is no more than n. If we have all $\bar{g}(t_1, t_2, m)$, then we can use the following dynamic programming recursion to calculate $\bar{G}(t, n)$,

$$\bar{G}(t, n) = \min_{\tau, m}\{\bar{g}(t, \tau, m) + \bar{G}(\tau + 1, n - m) \mid \tau = t, \ldots, T, m = 0, 1, \ldots, n\}, \quad (7)$$

with initial condition of $\bar{G}(T + 1, n) = 0$ for any $n \geq 0$, and boundary condition of $\bar{G}(t, n) = \infty$ for $n < 0$. The optimal solution is given by $\bar{G}(1, N)$.

Now we discuss how to calculate $\bar{g}(t_1, t_2, m)$ in the inner layer. First, for a sub-horizon with a single period, from definition we have

$$\bar{g}(t_1, t_1, m) = \begin{cases} 0 & \text{if } d_{t_1} \leq \bar{x}, \\ 0 & \text{if } \bar{x} < d_{t_1} \leq \bar{B} \text{ and } m \geq 1, \\ +\infty & \text{if } \bar{x} < d_{t_1} \leq \bar{B} \text{ and } m = 0, \\ +\infty & \text{if } \bar{B} < d_{t_1} \end{cases} \quad (8)$$

where $g(t_1, t_1, m) = +\infty$ indicates the violation of the definition of $g(t_1, t_1, m)$.

For $\bar{g}(t_1, t_2, m)$ with $t_1 < t_2$, the calculation involves another dynamic program. From Theorem 2 we know that for any τ, $\tau = t_1, t_1 + 1, \ldots, t_2 - 1$, x_τ is either $\bar{x}$ or $\bar{B}$ in an optimal solution. Among the periods $t_1, \ldots, \tau - 1$, if there are m' periods for $x_t = \bar{B}$, then there must be $\tau - t_1 - m'$ periods for $x_t = \bar{x}$. Thus the traffic delayed to period τ, denoted by $y_{\tau-1}(t_1, m')$, can be calculated as

$$y_{\tau-1}(t_1, m') = D_{t_1,\tau-1} - (m'\bar{B} + (\tau - t_1 - m')\bar{x}).$$

And we set $y_{t_1-1}(t_1, m') = 0$.

Now for $\tau = t_1, t_1 + 1, \ldots, T$, $m' = 0, 1, \ldots, N$, define $h(t_1, \tau, m')$ as the minimum delay penalty for periods t_1 to τ where (1) t_1 is the starting time of a sub-horizon which ends later than period τ, and (2) there are m' periods with $x_t = \bar{B}$ among periods $t_1, \ldots, \tau$. Comparing two possible cases of whether $x_\tau = \bar{B}$ or not, we have a dynamic programming recursion

$$h(t_1, \tau, m') = \min \begin{cases} h(t_1, \tau - 1, m') + f(y_{\tau-1}(t_1, m') + d_\tau - \bar{x}), \\ h(t_1, \tau - 1, m' - 1) + f(y_{\tau-1}(t_1, m' - 1) + d_\tau - \bar{B}). \end{cases} \quad (9)$$

We also have the boundary condition for (9) as $h(t_1, \tau, m') = +\infty$ for $y_\tau(t_1, m') \leq 0$, which enforces that a sub-horizon ends later than τ. For the initial condition with $\tau = t_1$, we have

$$h(t_1, t_1, 0) = \begin{cases} +\infty & \text{if } d_{t_1} \leq \bar{x}, \\ f(d_{t_1} - \bar{x}) & \text{if } d_{t_1} > \bar{x}, \end{cases} \quad (10)$$

$$h(t_1, t_1, 1) = \begin{cases} +\infty & \text{if } d_{t_1} \le \bar{B}, \\ f(d_{t_1} - \bar{B}) & \text{if } d_{t_1} > \bar{B}, \end{cases} \tag{11}$$

and $h(t_1, t_1, m') = +\infty$ for $m' \ge 2$, where the value of $+\infty$ indicates the violation of the definition of $h(t_1, t_1, m')$.

Then we can obtain $\bar{g}(t_1, t_2, m)$ based on $h(t_1, \tau, m')$. There are two possible cases: period t_2 is a peak period or not. For the case that period t_2 is not a peak period, we have

$$\bar{g}'(t_1, t_2, m) = \begin{cases} h(t_1, t_2 - 1, m) & \text{if } y_{t_2-1}(t_1, m) + d_{t_2} \le \bar{x}, \\ +\infty & \text{otherwise.} \end{cases}$$

For the case that period t_2 is a peak period, we have

$$\bar{g}''(t_1, t_2, m) = \begin{cases} h(t_1, t_2 - 1, m - 1) & \text{if } \bar{x} < y_{t_2-1}(t_1, m - 1) + d_{t_2} \le \bar{B}, \\ +\infty & \text{otherwise.} \end{cases}$$

Summarizing the above two cases, we have

$$\bar{g}(t_1, t_2, m) = \min\{\bar{g}'(t_1, t_2, m), \bar{g}''(t_1, t_2, m)\}. \tag{12}$$

Then the dynamic program given in (7) can be performed after first calculating all $\bar{g}(t_1, t_2, m)$'s from (8) to (12). We see that the number of $h(t_1, \tau, m')$'s is bounded by $O(T^2 N)$ and each of them can be calculated in constant time. Equation (7) can be calculated in $O(T^2 N^2)$ time after we have calculated all $\bar{g}(t_1, t_2, m)$'s. So the overall time complexity of the algorithm is in $O(T^2 N^2)$.

The implementation details and pseudo code for the dynamic program are given in Appendix 2.

5 Online implementation for the case of single-homed ISP

Up until this point, we have assumed that the traffic demand is known for each time period. In real time, we know that traffic demand fluctuates over time even though we have a very stable traffic demand pattern. In this section, we will discuss how the results for the offline case can be used to handle the real-time online case.

The difficulty in handling the online case is that the real traffic demand for a period is not known until the end of the period, but we have to (1) decide whether or not to make a period as a peak period at the beginning of the period while the real traffic demand for that period may only be known at the end of that period, and (2) design a scheme to implement the traffic regulation in real time. In particular, (1) is a planning problem that should be solved at the beginning of each time period, and (2) is a real-time scheduling problem for each packet. For the planning problem, we propose an online solution which uses the offline solution as a reference with an updating procedure to deal with the online errors. For the scheduling problem, we can apply an FIFO scheduling rule subject to the online planning decision.

5.1 Online planning

At the beginning of each period, we need to determine whether a period is to be a peak period or not. We model the online traffic by adding a random disruption to the offline traffic demand. In other words, the traffic demand pattern d_t for period t is an estimation for the traffic which is obtained from history data. For a period t, we assume that the real traffic is a random variable β_t, where $\beta_t = d_t + E_t$, and E_t is a random error from the estimation.

Before we present the solution to the online planning problem, we assume that the offline solution is available. To make the planning decision for a period $\hat{t}$, two components shall be considered: the traffic delay carried over from previous period $y_{\hat{t}-1}$, and the traffic demand estimation for the current period $d_{\hat{t}}$. Note that now $y_{\hat{t}-1}$ is the real traffic delay obtained from $\beta_1, \ldots, \beta_{\hat{t}-1}$. In our solution, if $y_{\hat{t}-1} + d_{\hat{t}} \leq \bar{x}$, then we know that period $\hat{t}$ is a non-peak period, regardless of the offline solution. For the case of $y_{\hat{t}-1} + d_{\hat{t}} > \bar{x}$:

- if period $\hat{t}$ is scheduled as a peak period in the offline schedule, then we set $\hat{t}$ as a peak period;
- if period $\hat{t}$ is not scheduled as a peak period, but there are fewer peak periods actually used than scheduled offline by period $\hat{t}$, then we set $\hat{t}$ as a peak period;
- for all other cases, we run an updated offline schedule starting from period $\hat{t}$, and follow this new offline schedule thereafter.

Note that for any of the above cases, the number of peak periods since the most recent starting time period shall be updated. For instance, in the case that offline schedule has to be updated, $\hat{t}$ will become the next starting time period. In our online policy, we may set period $\hat{t}$ as a peak period even if it has not been determined as a peak period in the original offline solution. Thus we can effectively handle an unexpected delay caused by a high traffic demand. This may lead to using one peak period earlier than originally scheduled, but the updating procedure guarantees that the total number of peak periods will not be overly used in the entire charging horizon, and thus the budget will not be exceeded.

For the implementation of the updating procedure, when an updated offline schedule is needed at period $\hat{t}$, we will apply the dynamic program in (7) which uses $t = \hat{t}$ as the starting time period rather than $t = 1$. In this process, we only need to re-calculate $\bar{g}(\hat{t}, t_2, m)$ based on the new traffic delay $y_{\hat{t}-1}$, which can be done in $O(TN)$ time, rather than all $\bar{g}(t_1, t_2, m)$ for all $t_1 \geq \hat{t}$. Similarly, we only need to calculate $\bar{G}(\hat{t}, n)$ in $O(TN)$ time, rather than $\bar{G}(t, n)$ for all $t \geq \hat{t}$. So the total involved calculation for the updating procedure can be done in $O(TN)$ time. This is important to satisfy the requirement of real time decision making for the online case.

5.2 Online packet scheduling

The implementation of our traffic regulation policy depends on a real-time online scheduling rule at the packet level. When each packet arrives, we need to decide whether to send it out immediately, or the put it in the buffer for a delay. We propose that a simple FIFO scheduling rule can be revised to support our traffic regulation policy.

Under our FIFO packet scheduling rule, a buffer queue is used to delay packets. When a packet arrives, it is put at the end of the buffer queue. Each time we decide to send a packet, the first packet in the buffer queue is sent out.

Suppose we are in a period t that starts from time s_t and ends at time e_t. The planned maximum traffic during this period is $\hat{x}_t$ which has been determined at the beginning of the period. Specifically, if the period is determined to be a peak period, then $\hat{x}_t = \bar{B}$, and if the period is determined not to be a peak period, then $\hat{x}_t = \bar{x}$.

At any time τ, $s_t \leq \tau < e_t$, we use v_τ to record the total traffic that has been sent out from time s_t to time τ. Suppose that a packet is just sent out at time τ. If the buffer queue is not empty, we check the first packet in the buffer. If the packet's length is l, then it takes $l/\bar{B}$ time to send the packet. We send the packet immediately if one of the following is true: (1) $\hat{x}_t = \bar{B}$; (2) $\hat{x}_t = \bar{x}$ and $v_\tau + l \leq \bar{x}$; or (3) $\hat{x}_t = \bar{x}$, $v_\tau + l > \bar{x}$, and $v_\tau + (e_t - \tau)\bar{B} \leq \bar{x}$. Otherwise, the packet will be delayed until time $e_t - (\bar{x} - v_\tau)/\bar{B}$ to ensure the packet is transmitted in a continuous way though it may be sent in two time periods.

The above packet scheduling rule sends packets as early as possible, while guaranteeing that the bandwidth usage will not exceed the capacity planned for that period.

6 Extension to dual-homed ISPs

In this section, we extend our model to the case where the customer is dual-homed to two ISPs both employing the percentile-based pricing policy. For any period, the customer needs to determine how to distribute the traffic to each ISP and how to delay some of the traffic. We will further discuss how to generalize the results to multi-homed K ISPs.

6.1 Some general results

For ISP i, $i = 1, 2$, assume that it has a bandwidth capacity $\bar{B}^{(i)}$, which allows for $N^{(i)}$ peak periods, and the charging point is $\bar{x}^{(i)}$. Recall that $N^{(i)}$ is solely determined by $\alpha^{(i)}$ given that ISP i takes an $\alpha^{(i)}$-percentile pricing policy. In many cases, we may have identical $N^{(i)}$ for both ISPs if they take the same α value. Here we discuss the general case. At any period t with traffic demand d_t, we need to decide the traffic distribution to ISP i, denoted by $x_t^{(i)}$, and the traffic to be delayed, denoted by y_t as before, with the flow conservation constraint $y_{t-1} + d_t = x_t^{(1)} + x_t^{(2)} + y_t$.

Before presenting our algorithm, we first use an example to illustrate the benefit of using dual-homed ISPs. The basic observation is that dual-homed ISPs may be able to tolerate more peak periods than a single ISP. Suppose that the number of charging periods is 100, and the traffic demand $d_t = t$ for $t = 1, \ldots, 100$. If a single ISP is used under a 95th percentile pricing policy with the charging volume being $\bar{x} = 95$, then the total delay penalty is 0. Now consider the case where two identical ISPs are used both with a 95th percentile pricing policy with $\bar{x}^{(1)} = \bar{x}^{(2)} = 45$. Then we can distribute traffic to each ISP in such a way that $x_t^{(i)} = d_t/2$ for $t = 1, \ldots, 90$ and $i = 1, 2$, $x_t^{(1)} = d_t$ for $t = 91, \ldots, 95$, $x_t^{(1)} = 0$ for $t = 96, \ldots, 100$, and $x_t^{(2)} = d_t - x^{(1)}$ for $t = 91, \ldots, 100$. In such a solution, again we have the total delay penalty being 0.

Assuming the cost is a linear function of the charging volume, then the user only needs to pay the bandwidth usage for $\bar{x}^{(1)} + \bar{x}^{(2)} = 90$ to the two ISPs, rather than the $\bar{x} = 95$ in the case of a single ISP. Therefore, the user can pay less for the same service quality if the user can distribute the traffic to two ISPs.

It might be inferred from the above example that a problem with dual-homed ISPs defined by $\{\bar{x}^{(i)}, \bar{B}^{(i)}, N^{(i)}\}$ can be transformed to an equivalent problem with a single virtual ISP where $\bar{x}^{(0)} = \bar{x}^{(1)} + \bar{x}^{(2)}$, $\bar{B}^{(0)} = \bar{B}^{(1)} + \bar{B}^{(2)}$ and $N^{(0)} = N^{(1)} + N^{(2)}$. Thus the problem can be easily solved. However, this is not always true. As shown in the following theorem, the single virtual ISP problem provides a lower bound of the total delay penalty for the dual-homed ISP problem, and they may be equivalent under special cases.

Theorem 3 *Let F_d be the minimum total delay penalty for a problem with dual-homed ISPs, and F_v be the minimum total delay penalty for the corresponding problem with a virtual single ISP, we have*

(1) $F_d \geq F_v$; *and*
(2) $F_d = F_v$ *if both $\bar{B}^{(1)}$ and $\bar{B}^{(2)}$ are infinite.*

Proof We use $\{x_t^{(1)}, x_t^{(2)}\}$ to denote a feasible solution to the dual-homed ISP problem, and $\{x_t^{(0)}\}$ to denote a feasible solution to the virtual single ISP problem. We will prove conclusion (1) by showing there is one $\{x_t^{(0)}\}$ for any given $\{x_t^{(1)}, x_t^{(2)}\}$, and further prove conclusion (2) by showing there is an $\{x_t^{(1)}, x_t^{(2)}\}$ for any given $\{x_t^{(0)}\}$ under the condition.

For any $\{x_t^{(1)}, x_t^{(2)}\}$, we can always have $\{x_t^{(0)}\}$ by letting $x_t^{(0)} = x_t^{(1)} + x_t^{(2)}$ which is feasible because (1) $x_t^{(0)} = x_t^{(1)} + x_t^{(2)} \leq \bar{B}^{(1)} + \bar{B}^{(2)} = \bar{B}^{(0)}$, and (2) at any period t, $x_t^{(0)}$ is a peak period only when at least one of $x_t^{(1)}$ and $x_t^{(2)}$ is a peak period, and thus the number of peak periods in $\{x_t^{(0)}\}$ is no more than the sum of number of peak periods in $\{x_t^{(1)}, x_t^{(2)}\}$. Therefore, we have conclusion (1) that $F_d \geq F_v$.

Now consider the case where both $\bar{B}^{(1)}$ and $\bar{B}^{(2)}$ are infinite. For any $\{x_t^{(0)}\}$, we can have $\{x_t^{(1)}, x_t^{(2)}\}$ as follows. If period t is not a peak period in $\{x_t^{(0)}\}$, i.e., $x_t^{(0)} \leq \bar{x}^{(0)}$, then arbitrarily set $x_t^{(1)}$ and $x_t^{(2)}$ such that $x_t^{(i)} \leq \bar{x}^{(i)}$ for $i = 1, 2$ and $x_t^{(1)} + x_t^{(1)} = x_t^{(0)}$. For the $N^{(0)}$ peak periods in $\{x_t^{(0)}\}$, let $N^{(1)}$ of them have $x_t^{(1)} = x^{(0)}$ and $x_t^{(2)} = 0$; and let $N^{(2)}$ other peak periods have $x_t^{(2)} = x^{(0)}$ and $x_t^{(1)} = 0$. So the total number of peak periods in $\{x_t^{(1)}, x_t^{(2)}\}$ is the same as $N^{(0)}$. Thus we have $F_d \leq F_v$ when both $\bar{B}^{(1)}$ and $\bar{B}^{(2)}$ are infinite. Combining with conclusion (1), we have conclusion (2) that $F_d = F_v$ if both $\bar{B}^{(1)}$ and $\bar{B}^{(2)}$ are infinite. □

The above theorem also shows that such a virtual single ISP provides more opportunity to reduce delay penalty. The reason is that the large virtual ISP allows more flexibility to handle more peak periods, which is generally known as the risk pooling effect (Shakkottai and Srikant 2005). In practice, however, there may not exist such a generous ISP who allows for so many peak periods. Most ISPs on the market use the same α in their pricing policy. As the previous example shows, in such a case, dual-homed ISP can help to improve the system performance.

6.2 Offline case

The offline case of dual-homed ISPs can be solved by following the similar idea for the case of a single ISP. We see that any feasible solution can be divided into a series of consecutive sub-horizons $[t_1, t_2]$ which are defined the same as the single ISP case. Furthermore, Theorem 2 for the case of single-homed ISP can be generalized as the following theorem.

Theorem 4 *There exists an optimal solution to the dual-ISP problem in which $y_t > 0$ implies that for each ISP i, either $x_t^{(i)} = \bar{x}^{(i)}$ or $x_t^{(i)} = \bar{B}^{(i)}$.*

Based on Theorem 4, we can revise the previous two-layer dynamic programming algorithm to deal with the dual-homed ISP problem. For the outer layer, let $\bar{g}(t_1, t_2, m^{(1)}, m^{(2)})$ be the minimum delay penalty for a sub-horizon $[t_1, t_2]$ when there are no more than $m^{(i)}$ peak periods being used for ISP i, $i = 1, 2$. Let $\bar{G}(t, n^{(1)}, n^{(2)})$ be the minimum total delay penalty from period t to T where (1) there is no delayed traffic carried over to period t, and (2) the number of peak periods is no more than $n^{(i)}$ for ISP i. Then we can use the following dynamic programming recursion to calculate $\bar{G}(t, n^{(1)}, n^{(2)})$,

$$\bar{G}(t, n^{(1)}, n^{(2)}) = \min_{\tau, m^{(1)}, m^{(2)}} \{\bar{g}(t, \tau, m^{(1)}, m^{(2)}) + \bar{G}(\tau + 1, n^{(1)} - m^{(1)}, n^{(2)} - m^{(2)}) \mid \tau = t, \ldots, T, m^{(i)} = 0, 1, \ldots, n^{(i)}\}, \tag{13}$$

with initial condition of $\bar{G}(T+1, n^{(1)}, n^{(2)}) = 0$ for $n^{(1)} \geq 0$ and $n^{(2)} \geq 0$, and boundary condition of $\bar{G}(t, n^{(1)}, n^{(2)}) = \infty$ for $n^{(1)} < 0$ or $n^{(2)} < 0$. The optimal solution is given by $\bar{G}(1, N^{(1)}, N^{(2)})$.

Now we discuss how to calculate $\bar{g}(t_1, t_2, m^{(1)}, m^{(2)})$ in the inner layer. First, for a sub-horizon with a single period, from definition we have $\bar{g}(t_1, t_1, m^{(1)}, m^{(2)}) = 0$ if any of the following conditions is satisfied: (1) $d_{t_1} \leq \bar{x}^{(1)} + \bar{x}^{(2)}$, or (2) $d_{t_1} \leq \bar{B}^{(1)} + \bar{x}^{(2)}$ and $\bar{m}^{(1)} \geq 1$, or (3) $d_{t_1} \leq \bar{x}^{(1)} + \bar{B}^{(2)}$ and $\bar{m}^{(2)} \geq 1$, or (4) $d_{t_1} \leq \bar{B}^{(1)} + \bar{B}^{(2)}$, $\bar{m}^{(1)} \geq 1$ and $\bar{m}^{(2)} \geq 1$; and $\bar{g}(t_1, t_1, m^{(1)}, m^{(2)}) = +\infty$ for other cases.

For $\bar{g}(t_1, t_2, m^{(1)}, m^{(2)})$ with $t_1 < t_2$, the calculation involves the inner-layer dynamic program. From Theorem 4 we know that for any τ, $\tau = t_1, t_1 + 1, \ldots, t_2 - 1$, $x_\tau^{(i)}$ is either $\bar{x}^{(i)}$ or $\bar{B}^{(i)}$ in an optimal solution. Among the periods $t_1, \ldots, \tau - 1$, if there are $l^{(i)}$ periods for $x_t^{(i)} = \bar{B}^{(i)}$, then there must be $\tau - t_1 - l^{(i)}$ periods for $x_t^{(i)} = \bar{x}^{(i)}$, and thus the traffic delay to period τ, denoted by $y_{\tau-1}(t_1, l^{(1)}, l^{(2)})$, can be calculated as

$$y_{\tau-1}(t_1, l^{(1)}, l^{(2)}) = D_{t_1, \tau-1} - \sum_{i=1}^{2} l^{(i)} \bar{B}^{(i)} - \sum_{i=1}^{2} (\tau - t_1 - l^{(i)}) \bar{x}^{(i)}.$$

Now for $\tau = t_1, t_1 + 1, \ldots, T$, $l^{(i)} = 0, 1, \ldots, N^{(i)}$, define $h(t_1, \tau, l^{(1)}, l^{(2)})$ to be the minimum delay penalty for periods t_1 to τ where (1) t_1 is the start time of a sub-horizon which ends later than period τ, and (2) there are $l^{(i)}$ periods with $x_t^{(i)} = \bar{B}^{(i)}$

among periods $t_1, \ldots, \tau$. Comparing four possible cases of whether $x_\tau^{(i)} = \bar{B}^{(i)}$ or not, we have a dynamic programming recursion

$$h(t_1, \tau, l^{(1)}, l^{(2)}) = \min \begin{cases} f(y_{\tau-1}(t_1, l^{(1)}, l^{(2)}) + d_\tau - \bar{x}^{(1)} - \bar{x}^{(2)}) \\ \quad + h(t_1, \tau - 1, l^{(1)}, l^{(2)}), \\ f(y_{\tau-1}(t_1, l^{(1)} - 1, l^{(2)}) + d_\tau - \bar{B}^{(1)} - \bar{x}^{(2)})) \\ \quad + h(t_1, \tau - 1, l^{(1)} - 1, l^{(2)}), \\ f(y_{\tau-1}(t_1, l^{(1)}, l^{(2)} - 1) + d_\tau - \bar{x}^{(1)} - \bar{B}^{(2)})) \\ \quad + h(t_1, \tau - 1, l^{(1)}, l^{(2)} - 1), \\ f(y_{\tau-1}(t_1, l^{(1)} - 1, l^{(2)} - 1) + d_\tau - \bar{B}^{(1)} - \bar{B}^{(2)})) \\ \quad + h(t_1, \tau - 1, l^{(1)} - 1, l^{(2)} - 1). \end{cases}$$

We also have the boundary condition for the above equation as $h(t_1, \tau, l^{(1)}, l^{(2)}) = +\infty$ for $y_\tau(t_1, l^{(1)}, l^{(2)}) \leq 0$, which enforces that a sub-horizon ends later than τ. For the initial condition with $\tau = t_1$, we have

$$h(t_1, t_1, 0, 0) = \begin{cases} +\infty & \text{if } d_{t_1} \leq \bar{x}^{(1)} + \bar{x}^{(2)}, \\ f(d_{t_1} - \bar{x}^{(1)} - \bar{x}^{(2)}) & \text{otherwise,} \end{cases}$$

$$h(t_1, t_1, 1, 0) = \begin{cases} +\infty & \text{if } d_{t_1} \leq \bar{B}^{(1)} + \bar{x}^{(2)}, \\ f(d_{t_1} - \bar{x}^{(1)} - \bar{x}^{(2)}) & \text{otherwise,} \end{cases}$$

$$h(t_1, t_1, 0, 1) = \begin{cases} +\infty & \text{if } d_{t_1} \leq \bar{x}^{(1)} + \bar{B}^{(2)}, \\ f(d_{t_1} - \bar{x}^{(1)} - \bar{B}^{(2)}) & \text{otherwise,} \end{cases}$$

$$h(t_1, t_1, 1, 1) = \begin{cases} +\infty & \text{if } d_{t_1} \leq \bar{B}^{(1)} + \bar{B}^{(2)}, \\ f(d_{t_1} - \bar{B}^{(1)} - \bar{B}^{(2)}) & \text{otherwise,} \end{cases}$$

and $f(t_1, t_1, l^{(1)}, l^{(2)}) = +\infty$ for $l^{(i)} \geq 2$, $i = 1, 2$, where the value of $+\infty$ indicates the violation of the definition of $h(t_1, t_1, l^{(1)}, l^{(2)})$.

Then we can obtain $\bar{g}(t_1, t_2, m^{(1)}, m^{(2)})$ based on $h(t_1, \tau, l^{(1)}, l^{(2)})$. There are four possible cases with respect to period t_2 being a peak period or not.

Case 1 For period t_2 being not a peak period for either ISP, if $y_{t_2-1}(t_1, m^{(1)}, m^{(2)}) + d_{t_2} \leq \bar{x}^{(1)} + \bar{x}^{(2)}$, then

$$\bar{g}_1(t_1, t_2, m^{(1)}, m^{(2)}) = h(t_1, t_2 - 1, m^{(1)}, m^{(2)}),$$

otherwise, $\bar{g}_1(t_1, t_2, m^{(1)}, m^{(2)}) + \infty$.

Case 2 For period t_2 being a peak period for ISP 1 only, if $y_{t_2-1}(t_1, m^{(1)}-1, m^{(2)}) + d_{t_2} \le \bar{B}^{(1)} + \bar{x}^{(2)}$, then

$$\bar{g}_2(t_1, t_2, m^{(1)}, m^{(2)}) = h(t_1, t_2-1, m^{(1)}-1, m^{(2)}),$$

otherwise, $\bar{g}_2(t_1, t_2, m^{(1)}, m^{(2)}) = +\infty$.

Case 3 For period t_2 being a peak period for ISP 2 only, if $y_{t_2-1}(t_1, m^{(1)}, m^{(2)}-1) + d_{t_2} \le \bar{x}^{(1)} + \bar{B}^{(2)}$, then

$$\bar{g}_3(t_1, t_2, m^{(1)}, m^{(2)}) = h(t_1, t_2-1, m^{(1)}, m^{(2)}-1),$$

otherwise, $\bar{g}_3(t_1, t_2, m^{(1)}, m^{(2)}) = +\infty$.

Case 4 For period t_2 being a peak period for both ISPs, if $y_{t_2-1}(t_1, m^{(1)}-1, m^{(2)}-1) + d_{t_2} \le \bar{B}^{(1)} + \bar{B}^{(2)}$, then

$$\bar{g}_4(t_1, t_2, m^{(1)}, m^{(2)}) = h(t_1, t_2-1, m^{(1)}-1, m^{(2)}-1),$$

otherwise, $\bar{g}_4(t_1, t_2, m^{(1)}, m^{(2)}) = +\infty$.

Summarizing the above four cases, we have

$$\bar{g}(t_1, t_2, m^{(1)}, m^{(2)}) = \min\{\bar{g}_i(t_1, t_2, m) \mid i = 1, 2, 3, 4\}. \tag{14}$$

Again, the time complexity of the above algorithm is dominated by the outer layer dynamic program given in (13) where the number of states is bounded by $O(TN^{(1)}N^{(2)})$ and each state can be evaluated in $O(TN^{(1)}N^{(2)})$ time. So the overall time complexity of the algorithm is in $O(T^2(N^{(1)})^2(N^{(2)})^2)$.

We will omit the implementation details for the above dynamic program as it is similar to the single ISP case.

6.3 Online planning

We now discuss the online problem for the dual-homed ISPs.

For the online planning problem, we need to determine whether a peak period $\hat{t}$ should be incurred to an ISP given $y_{\hat{t}-1}$ and $d_{\hat{t}}$. Similar to the single ISP case, we propose the following scheme:

- Case 1: if $y_{\hat{t}-1} + d_{\hat{t}} \le \bar{x}^{(1)} + \bar{x}^{(2)}$, then each ISP has a non-peak period at $\hat{t}$; otherwise,
- Case 2: if $y_{\hat{t}-1} + d_{\hat{t}} \le \bar{B}^{(1)} + \bar{x}^{(2)}$ and (i) ISP 1 has a planned peak period at $\hat{t}$ in the offline schedule or (ii) there are fewer peak periods incurred to ISP 1 than offline scheduled, then $\hat{t}$ is a peak period to ISP 1; otherwise,
- Case 3: if $y_{\hat{t}-1} + d_{\hat{t}} \le \bar{x}^{(1)} + \bar{B}^{(2)}$ and (i) ISP 2 has a planned peak period at $\hat{t}$ in the offline schedule or (ii) there are fewer peak periods incurred to ISP 2 than offline scheduled, then $\hat{t}$ is a peak period to ISP 2; otherwise,

- Case 4: if $y_{\hat{t}-1} + d_{\hat{t}} \leq \bar{B}^{(1)} + \bar{B}^{(2)}$ and both ISPs have a planned peak period at $\hat{t}$ in the offline schedule, then $\hat{t}$ is a peak period to ISP 1; otherwise,
- Case 5: we run an updated offline schedule for period $\hat{t}$.

For the aim of load balancing between the two ISPs, we may rotate the sequence of Case 2 and Case 3 period by period in the above scheme.

Similar to the case of a single ISP, the time complexity of Case 5 for obtaining an updated schedule is $O(TN^{(1)}N^{(2)})$ because we only need to recalculate $\bar{G}(\hat{t}, n^{(1)}, n^{(2)})$ which enumerates $\bar{g}(\hat{t}, \tau, m^{(1)}, m^{(2)})$ over $\tau, m^{(1)}, m^{(2)}$, and all $\bar{g}(\hat{t}, \tau, m^{(1)}, m^{(2)})$'s can be obtained in $O(TN^{(1)}N^{(2)})$ time.

6.4 Online packet scheduling

For online packet scheduling, an FIFO rule is modified as follows. Given a packet to be sent out, first we need to decide which ISP to use. For load balancing purpose, we propose a random selection scheme. We first choose one ISP based on certain probability; if it is determined that this ISP is to delay the packet, then we try the other ISP.

The ISP selection probability is proportional to its planned capacity for that period. For a period t that starts from time s_t and ends at e_t, we use $\hat{x}^{(i)}$ to denote the maximum planned traffic for ISP i during a period t. We know that $\hat{x}^{(i)} \in \{\bar{x}^{(i)}, \bar{B}^{(i)}\}$, depending on whether period t is determined as a peak period in the online planning. Then the selection probability for ISP i would be $\hat{x}^{(i)}/(\hat{x}^{(1)} + \hat{x}^{(2)})$.

After an ISP i is selected, we can try to use that ISP to send the packet by following the packet scheduling scheme for a single ISP. The details are omitted.

6.5 Multi-homed ISPs

The above approach can be generalized to multi-homed case with K available ISPs where we develop a dynamic program by defining $\bar{G}(t, n^{(1)}, \ldots, n^{(K)})$. However, the number of states will become exponential with K, and thus such a dynamic programming approach becomes inefficient. In fact, we can show below that the problem becomes intractable with an arbitrary K.

Theorem 5 *The decision version of the multi-homed problem with K ISPs is NP-complete in the strong sense.*

Proof We prove the theorem by a reduction from 3-PARTITION problem, a known NP-complete problem in the strong sense.

3-PARTITION Problem: *Given a set A of* $3n$ *integers* $\{a_1, a_2, \ldots, a_{3n}\}$ *and an integer* b, *where* $b/4 < a_i < b/2$ *and* $\sum_{i=1}^{3n} a_i = nb$, *we ask whether A can be partitioned into n disjoint subsets* $A_1, \ldots, A_n$ *such that* $\sum_{a_i \in A_t} = b$ *for* $t = 1, \ldots, n$.

Given an instance of the 3-PARTITION problem, we can construct an instance of the decision version of the multi-homed problem with K ISPs as follows. Let there

be $K = 3n$ ISPs where each ISP i has a charging point $\bar{x}^{(i)} = b$, allowable peak period $N^{(i)} = 1$, and capacity $\bar{B}^{(i)} = b + a_i$. Let there be $T = n$ time periods, where the traffic demand $d_t = (3n + 1)b$ for $t = 1, \ldots, n$. Obviously the above problem construction procedure takes polynomial time. The question is to ask whether there exists a solution to such a multi-homed problem with zero delay penalty.

Suppose that the 3-PARTITION problem has a YES solution $\{A_1, \ldots, A_n\}$. Then we can have a solution to the above multi-homed problem where in each period t all ISPs are first assigned by a traffic up to the charging point b, then for the three ISPs corresponding to the three integers a_i in A_t, each is assigned by an extra traffic of a_i. In such a solution, there incurs only one peak period to each ISP, and there is no traffic delay at any time period.

Suppose that there exists a YES solution to the above multi-homed problem with zero delay penalty. In such a solution, for any period t there must be peak periods incurred to exactly three ISPs because (1) each ISP cannot have one peak period, and (2) the traffic demand at each period requires peak periods incurred to at least three ISPs. Therefore, we have a YES solution to the 3-PARTITION problem where the three ISPs with their peak periods incurred in the same time period t correspond to the three integers in a subset A_t. □

7 Problem decomposition

When the charging horizon is long, for example, one month or even longer, the above two dynamic programming algorithms become inefficient with respect to both computation time and memory usage. In this section, we discuss how the problem with a large charging horizon can be decomposed into a series of smaller problems with a small charging horizon. We only discuss the case with the capacity constraint. The easier case without the capacity constraint can be handled in the same way.

We first observe that in practice the traffic demand actually has some periodic pattern over days. In particular, the daily traffic demand is high during day time, and low after midnight (Fukuda et al. 2003). If the traffic demand is always below a given $\bar{x}$ during certain time for each day, then the problem with multiple days can be decomposed into multiple problems each for a single day. For example, if we know that the traffic demand is lower than $\bar{x}$ during 3am–4am each day, then we can fairly assume that in an optimal solution to the problem with long time periods, there is no traffic demand being delayed to 4am. This implies that the optimal schedule for one day (starting from 4am) is independent of the optimal schedule for the following days. Therefore, we can solve the problem on a daily base.

If each day has the same traffic demand pattern, then the problem is trivially solved by such a decomposition because we only need to solve a single-day problem and then repeat the single-day solution day by day. In practice, however, we often see that the daily traffic demand varies. For example, the traffic demand is low on weekend, and high on weekdays. The immediate consequence of such a phenomenon is that each day needs a different schedule. Furthermore, we need to assign different number of peak periods to different days. There are in total $N = (1 - \alpha)T$ peak periods in a charging horizon, but it does not necessarily mean that we will evenly allocate these

N peak periods to different days. For example, we may allocate more peak periods to weekdays than weekends. Therefore, we face the problem of how to determine the number of peak periods for each day for the problem decomposition.

Suppose that the whole charging horizon contains m days, and each day contains $\hat{T}$ periods. If it is decided that there are N_s peak periods for day s, $s = 1, \ldots, m$, and $N_1 + \ldots + N_m = N$, then the entire problem is decomposed into m smaller problems each with N_s peak periods; each smaller problem s can be solved with $\hat{T}$ total periods and N_s peak periods; and a delay penalty $\hat{G}_s(N_s)$ can be obtained. Note that we can obtain $\hat{G}_s(N_s)$ for different values of N_s in a single run of the dynamic program (7). The only problem left is how to determine the values of N_s.

We now propose another dynamic programming algorithm to optimally assign the number of peak periods. Let $F(s, \theta)$ be the minimum total delay penalty from day s to the end of the charging horizon, given that there are θ peak periods allowed starting from day s. Then we have a dynamic programming recursion as follows.

$$F(s, \theta) = \min_{N_s} \{\hat{G}_s(N_s) + F(s+1, \theta - N_s) | 0 \le N_s \le \theta\}, \tag{15}$$

where the initial condition is $F(m, \theta) = \hat{G}_m(\theta)$. The optimal allocation of the peak periods can then be obtained from $F(1, N)$.

The time complexity for such a problem decomposition is analyzed as follows. All $\hat{G}_s(N_s)$'s can be calculated in $O(m\hat{T}^2N^2)$ time, then the dynamic program (15) is performed in (mN^2) time. So the overall time complexity is in $O(m\hat{T}^2N^2)$ time.

For the case of dual-homed ISPs, the problem decomposition can be performed as follows. Let $F(s, \theta^{(1)}, \theta^{(2)})$ be the minimum total delay penalty from day s to the end of the charging horizon, given that there are $\theta^{(i)}$ peak periods allowed by ISP i starting from day s. Then we have a dynamic programming recursion as follows.

$$F(s, \theta^{(1)}, \theta^{(2)}) = \min_{N_s^{(1)}, N_s^{(2)}} \{\hat{G}_s(N_s^{(1)}, N_s^{(2)}) + F(s+1, \theta^{(1)} - N_i^{(1)}, \theta^{(2)} - N_i^{(2)}) \mid$$

$$0 \le N_s^{(i)} \le \theta^{(i)}\},$$

where the initial condition is $F(m, \theta^{(1)}, \theta^{(2)}) = \hat{G}_m(\theta^{(1)}, \theta^{(2)})$. The optimal allocation of the peak periods can then be obtained from $F(1, N^{(1)}, N^{(2)})$. Similarly, the time complexity for the decomposition would be in $O(m\hat{T}^2(N^{(1)})^2(N^{(2)})^2)$.

8 Experimental results

In this section, we report the experimental results. The purposes of the experiments are (1) to understand the tradeoff between the cost and service quality, (2) to study the benefit of dual-homed ISPs, and (3) to investigate the effectiveness of our proposed policies for the online case.

Our experiments are conducted based on the data from an Internet trace (NLANR and NLANR PMA 2005). Reading in the raw data from the trace file, we make a transformation so that a series of traffic demand is obtained for every five minutes during a single day period, which implies that $T = 288$ and $N = 14$. In the data

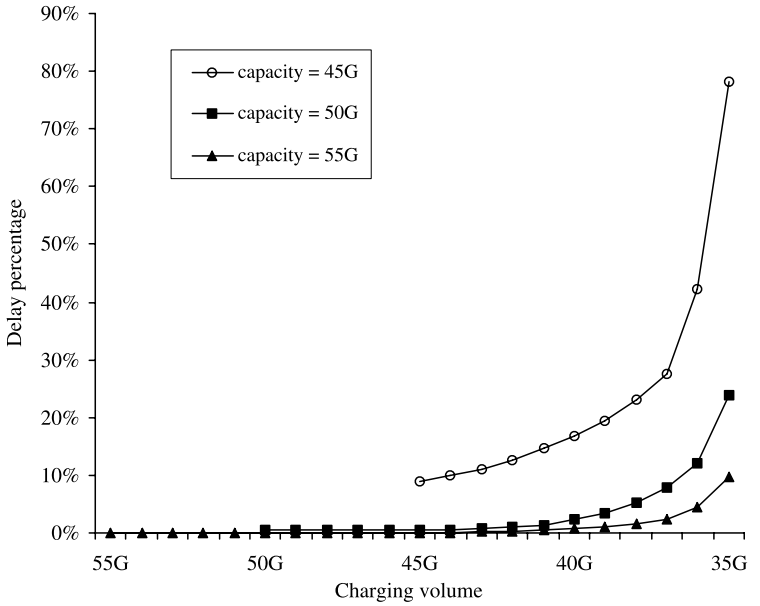

Fig. 1 Tradeoff between cost and delay for a single ISP

set, the maximum traffic demand for one period is $53.2GB$, the minimum demand is $9.8GB$, and the total traffic demand over the entire day is $5658.4GB$. We also note that the 14-th largest traffic demand is $46.2GB$, the value that would be used as the charging volume if the user does not take any action under a large enough bandwidth capacity. We use $f(y) = y$ as the delay penalty function for delayed traffic y.

We first study the tradeoff between the cost and service quality. For the given traffic demand series, we have investigated three different cases with capacity $\bar{B} = 55G$, $50G$, and $45G$, respectively (for simplicity, we shorten GB as G in all figures). For each $\bar{B}$, different values of $\bar{x}$ are used. For the service quality evaluation, in order to have a more meaningful result, we report the average delay penalty rather than the absolute total delay penalty that is used in the model. Specifically, we define the average delay penalty as the ratio of total delay penalty over the total traffic demand, a value that can be interpreted as the percentage of traffic demand that is delayed by one time period. The results are given in Fig. 1.

In Fig. 1, we see that the delay percentage increases when the charging volume $\bar{x}$ decreases for all $\bar{B}$ values, indicating the deteriorative service quality (in terms of delay) caused by cost saving. For example, when the capacity $\bar{B} = 55G$, the delay percentage is below 1% with $\bar{x} = 40G$. Note that if the user does not take any action, the charging volume is $46.2G$ with zero delay because the maximum traffic is $53.2G$ and the 14-th largest traffic demand is $46.2G$. Assuming the pricing is approximately linear with the charging volume, this implies that we can reduce the cost by about $1 - 40/46.2 \cong 13\%$ with the delay performance being worse by no more than 1%. Similarly, we see that the delay percentage is below 5% with $\bar{x} = 36G$, implying that

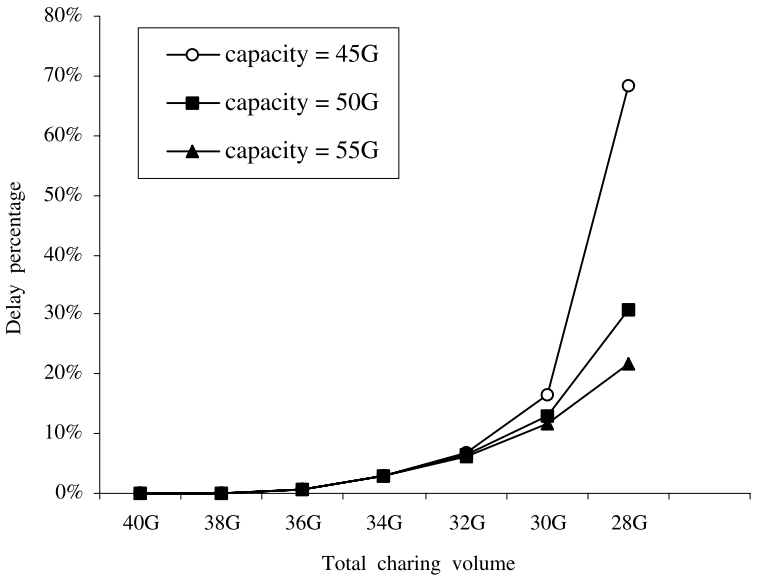

Fig. 2 Tradeoff between cost and delay for dual-homed ISPs

we can reduce the cost by about $1 - 36/46.2 \cong 22\%$ with the delay performance being worse by no more than 5%. This shows the benefit of traffic regulation.

We also notice that the delay performance gets worse with the decrease of capacity $\bar{B}$. For a given charging volume $\bar{x}$, a large capacity can help to reduce delayed traffic. For example, at $\bar{x} = 40G$, the delay is below 1% with $\bar{B} = 55G$, about 2% with $\bar{B} = 50G$, and about 16% with $\bar{B} = 45G$. If the pricing scheme solely depends on the percentile charging volume, or the capacity cost is a one-time expense, then it is better to set up a large capacity in order to reduce delay.

Now we show the case of using dual-homed ISPs in Fig. 2. We assume that the two ISPs are with the same maximum capacity, and given the same charging volume $\bar{x}^{(1)} = \bar{x}^{(2)}$. In the figure, the horizontal axis is the sum of two charging volumes $\bar{x}^{(1)} + \bar{x}^{(2)}$; for example, the total charging volume $40G$ means $\bar{x}^{(1)} = \bar{x}^{(2)} = 20G$. The total charging volume indicates the cost that the user needs to pay under the assumption of a linear cost function of the charging volume.

While Fig. 2 clearly shows the tradeoff between delay and cost for dual-homed ISPs, it is more interesting to compare it with the case of a single ISP under the same charging volume. For example, at the charging volume $35G$, the user has started to experience significant delay under a single ISP, but under the dual-homed ISP, the delay is still very small until the total charging volume is smaller than $30G$. This shows the benefit of using dual-homed ISPs.

In Fig. 2, we assume a given charging volume is evenly shared by two ISPs, which seems to be an intuitively reasonable scheme because these two ISPs have identical capacity and use the same percentile charging policy. However, as we are to present in Fig. 3, this is not the best allocation.

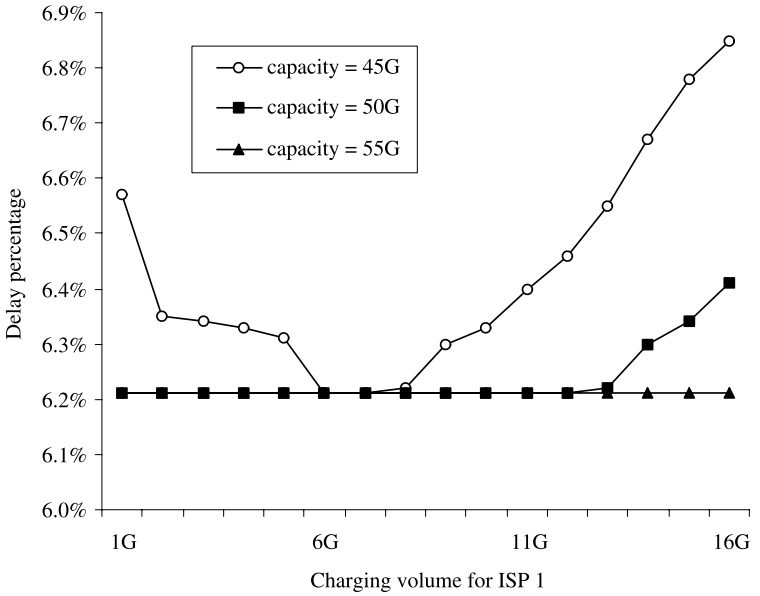

Fig. 3 Charging volume allocation for dual-homed ISPs

In Fig. 3, we assume the total charging volume is $32G$ which is to be allocated to two ISPs, where we use the horizontal axis to represent the charging volume for ISP 1. The figure shows that when the capacity is small, the even allocation, $16G$ to each ISP, has a larger delay percentage than other cases. It is better to set a higher charging volume to one ISP and a lower charging volume to another ISP. While this is not an intuitive result, it can be explained as by an example with $\bar{B}_1 = \bar{B}_2 = 45G$. Suppose at a period t we have $y_{t-1} + d_t = 63G$. If the charging volume for each ISP is evenly set as $16G$, then both ISPs have to be with peak periods in order not to delay any traffic to period $t+1$; if $\bar{x}^{(1)} = 13G$, $\bar{x}^{(2)} = 19G$, then we can assign $45G$ to ISP 1 and $18G$ to ISP 2 without any traffic delay to period $t+1$. However, under such allocation, there is only one peak period, which is occurred to ISP 1. Therefore, one peak period is saved for later periods to reduce delay. This example shows that an uneven assignment of charging volume can bring more flexibility to reduce traffic delay.

There are other observations for the charging volume allocation. First, while an uneven allocation may be better than an even allocation, the benefit may not be proportional to the allocation difference between the ISPs. When the difference becomes too large, the traffic delay starts to increase again. Second, the delay curve is not convex to the allocation to one ISP, implying that it is not easy to find the best allocation. Finally, the capacity plays an important role in the allocation. When the capacity is very large, say $55G$, different allocation virtually makes no difference.

We now test our online policy. As we have mentioned, we model the online traffic demand as the offline estimation plus a random error. In our computation, we generate the random error from a normal distribution $N(0, \sigma^2)$ with the standard deviation

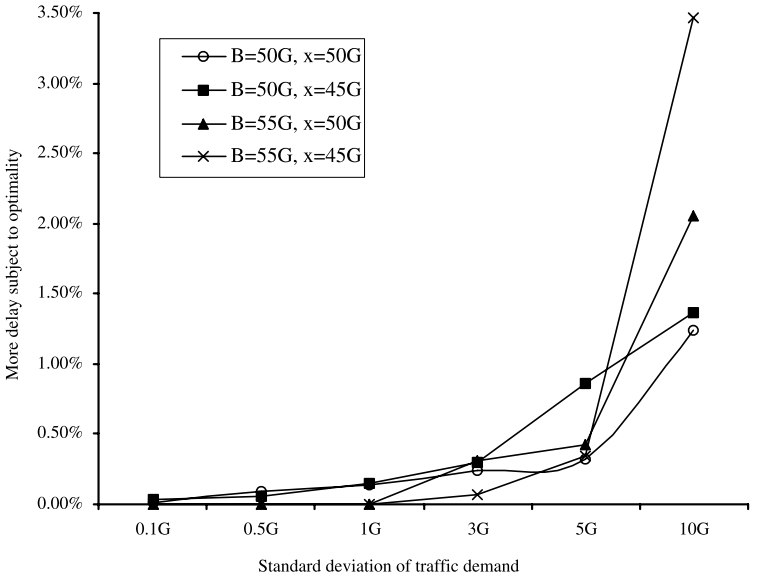

Fig. 4 Performance of the online problem for a single ISP

$\sigma = 0.1G, 0.5G, 1G, 3G, 5G$, and $10G$. To evaluate the effectiveness of our policy, for any given $\bar{B}$ and $\bar{x}$, we compare the average delay obtained from our online policy with the ideal optimal solution if the real traffic demand is known in advance. Note that such an ideal solution represents the lower bound of the average delay for any online policy.

We report the comparison in Fig. 4 for the case of a single ISP, which shows the extra delay in our online policy subject to the ideal solution. We see that the difference is below 1% even when the standard deviation of the real traffic demand $\sigma = 5G$, which is about 10% of the maximum traffic estimation. This evidences the effectiveness of our online policy. Not surprisingly, the performance of our online policy becomes worse when σ increases. When σ is larger than $10G$, which is about 20% of the maximum traffic estimation, our online policy may cause a large traffic delay, and become not effective. These cases, however, have violated our assumption that the traffic demand is relatively stable subject to a known pattern.

We also find in Fig. 4 that the performance of our online policy is not sensitive to the capacity $\bar{B}$ and predetermined charging volume $\bar{x}$ when the deviation σ is small. This indicates the robustness of our policy in certain sense. When σ is large, our online policy will perform relatively better under a compact capacity $\bar{B} = 50G$. When $\bar{B}$ is large, it brings more optimization room for the ideal solution, thus our online policy may become less competitive.

At last, we report the online performance under dual-homed ISPs, which is shown in Fig. 5. Similar to the single ISP case, we see that our online policy is very stable subject to a small traffic deviation σ, e.g., $\sigma \leq 3G$. However, when the traffic deviation is large, the online performance for the dual-homed ISP becomes worse

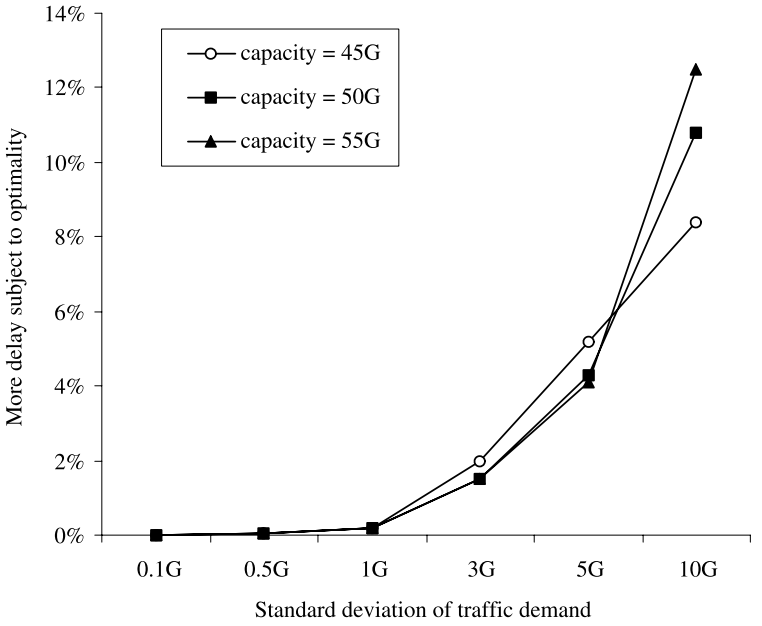

Fig. 5 Performance of the online problem for dual-homed ISPs

more quickly than the single ISP case, which brings more challenging problems for handling dual-homed ISPs.

9 Conclusion

In this paper, we have discussed how a customer can balance cost and packet buffer delay facing a percentile-based pricing scheme. The main contribution is about how a customer can save the Internet accessing cost without too much sacrifice of the service quality by strategically delaying some traffic demand. We optimally solve the problem for the offline case, and develop a dynamic scheduling policy for the online case which uses the offline solution as a reference. We also extend our results to the case of dual-homed ISPs.

We believe that the work can be further extended in several different ways. In particular for the online case, more sophisticated policy may be desired for the case that the traffic demand does not have a stable pattern, or the case with multi-homed ISPs.

Appendix 1: Algorithm implementation for the case without capacity constraint

We now discuss the details of the implementation of the dynamic program given in (6). Besides the total delay penalty that can be obtained from $G(1, N)$, the implementation also needs a backward tracking procedure to find the optimal traffic for each

Fig. 6 Algorithm for the case without capacity constraint

```
begin
//calculating g(t1, t2)
for t1=1 to T do
  g(t1, t1) = 0; z(t1, t1) = 0;
  for t2 = t1 + 1 to T do
    calculate z(t1, t2), g(t1, t2), and δ(t1, t2)
  end for//t2
end for //t1

//calculating G(t, n)
initialize G(T + 1, n) = 0 for n = 0, 1, ..., N
for t = T downto 1 do
  for n = 1 to N do
    G(t, n) = ∞
    for τ = t to T do
      if G(t, n) > g(t, τ) + G(τ + 1, n − δ(t, τ)) then
        G(t, n) = g(t, τ) + G(τ + 1, n − δ(t, τ))
        S(t, n) = τ
      end if
    end for //τ
  end for //n
end for //t

//solution output
t = 1; n = N;
while t < T do
  τ = S(t, n)
  if δ(t, τ) = 1 then period τ is peak period
  n = n − δ(t, τ), t = τ+1
end while
end
```

period. In particular, we need to know which periods are peak periods. This can be done by recording how each $G(t, n)$ is obtained from (6). Specifically, we can use a solution table $S(t, n)$ to denote the optimal τ that leads to the minimum $G(t, n)$, i.e., $S(t, n) = \tau^*$ where τ^* is the last period of a sub-horizon in the optimal solution, and τ^* satisfies

$$g(t, \tau^*) + G(\tau + 1, n - \delta(t, \tau^*)) = \min_{\tau}\{g(t, \tau) + G(\tau + 1, n - \delta(t, \tau)) \mid \tau = t, t + 1, \ldots, T\}.$$

After calculating $G(1, N)$, we can determine all peak periods from $t = 1$ based on the information given in $S(t, n)$. We start from $t = 1$ and $n = N$, repeatedly find period $\tau = S(t, n)$ that leads to $G(t, n)$, until the end of the charging horizon. These periods are the last periods in all sub-horizons and they divide the charging horizon

Fig. 7 Algorithm for the case with capacity constraint

begin
//calculate $y_{\tau-1}(t_1, m)$
for $t_1 = 1$ **to** T **do**
  for $\tau = t_1$+1 **to** T **do**
    for $m = 0$ **to** N **do**
      $y_{\tau-1}(t_1, m) = D_{t_1,\tau-1} - (m\bar{B} + (\tau - t_1 - m)\bar{x})$
    end for //m
  end for //τ
end for //t_1

//calculate $h(t_1, \tau, m')$
for $t_1 = 1$ **to** T **do**
  for $\tau = t_1$ **to** T **do**
    for $m' = 0$ **to** N **do**
      calculate $h(t_1, \tau, m')$ and $\bar{h}(t_1, \tau, m')$
    end for //m'
  end for //τ
end for //t_1

//calculate $\bar{g}(t_1, t_2, m)$
for $t_1 = 1$ **to** T **do**
  for $t_2 = t_1$ **to** T **do**
    for $m = 0$ **to** N **do**
      calculate $\bar{g}(t_1, t_2, m)$ and $\bar{S}(t_1, t_2, m)$
    end for //m
  end for //t_2
end for //t_1

//calculate $\bar{G}(t, n)$
for $t = T$ **downto** 1 **do**
  for $n = 0$ **to** N **do**
    calculate $G(t, n)$, $\tau^*(t, n)$ and $m^*(t, n)$
  end for //n
end for //t

//output Solution
$t = 1, n = N$
while $t < T$ **do**
  $\tau = \tau^*(t, n), m = m^*(t, n)$
  output peak periods in $\bar{S}(\tau, m)$
  $t = \tau + 1, n = n - m$
end while
end

into sub-horizons. According to the description in Sect. 4.1, only these periods can be peak periods. By checking $\delta(t,\tau)$ for $\tau = S(t,n)$, we can find out if τ is a peak period or not. The overall algorithm is given in Fig. 6.

Appendix 2: Algorithm implementation for the case with capacity constraint

The implementation of the algorithm for the case with capacity constraint needs the following steps. First, calculate all $y_{\tau-1}(t,m')$ for $t = 1,\ldots,T$, $\tau = t,\ldots,T$ and $m' = 0,\ldots,N$; second, calculate all $h(t_1,\tau,m')$ for $t_1 = 1,\ldots,T$, $\tau = t_1,\ldots,T$ and $m' = 0,\ldots,N$; third, calculate all $\bar{g}(t_1,t_2,m)$ for $t_1 = 1,\ldots,T$, $t_2 = t_1,\ldots,T$, and $m = 0,\ldots,N$; and finally, calculate $\bar{G}(t,n)$ from $\bar{g}(t_1,t_2,m)$, obtain the minimum total delay penalty from $\bar{G}(1,N)$, and output all peak periods.

We also need a backward tracking procedure to find those peak periods in an optimal solution. For any $\bar{g}(t_1,t_2,m)$, we define a set $\bar{S}(t_1,t_2,m)$ that records the peak periods for $\bar{g}(t_1,t_2,m)$. Then in calculating $\bar{G}(t,n)$ from (7), we use $\tau^*(t,n)$ and $m^*(t,n)$ to denote the τ and m that minimize $\bar{G}(t,n)$, i.e.,

$$\begin{aligned}&\bar{g}(t,\tau^*(t,n),m^*(t,n)) + \bar{G}(\tau^*(t,n)+1, n-m^*(t,n))\\ &\quad = \min_{\tau,m}\{\bar{g}(t,\tau,m) + \bar{G}(\tau+1,n-m) \mid \tau = t,\ldots,T, m = 0,1,\ldots,n\}.\end{aligned}$$

After calculating $\bar{G}(1,N)$, we can determine all peak periods starting from $t = 1$ and $n = N$ where $\bar{S}(\tau^*(t,n),m^*(t,n))$ gives all peak periods during t and $\tau^*(t,n)$; then we update $t = \tau^*(t,n)+1$ and $n = n - m^*(t,n)$; repeat the above steps until $t > T$.

In order to get $\bar{S}(t_1,t_2,m)$, we need to maintain a three-dimensional array $\bar{h}(t_1,\tau,m')$ associated with the calculation of each $h(t_1,\tau,m')$, indicating whether period τ should be a peak period. When $h(t_1,\tau,m')$ is obtained from the second term of (9) or (11), we set $\bar{h}(t_1,\tau,m') = 1$; for other cases, we set $\bar{h}(t_1,\tau,m') = 0$. After obtaining $\bar{g}(t_1,t_2,m)$, we will be able to get the peak periods for $\bar{g}(t_1,t_2,m)$ from $\bar{h}(t_1,\tau,m')$ as follows. Let $\tau = t_2$ and $m' = m$. If $\bar{g}(t_1,t_2,m)$ is equal to the first term of (12), period τ is not peak; otherwise, period τ is not peak, and let $m' = m' - 1$. Then repeat the following until $\tau = t_1$: Let $\tau = \tau - 1$; if $h(t_1,\tau,m') = 1$, period τ is peak; let $m' = m' - h(t_1,\tau,m')$.

The detailed steps for the algorithm is described in Fig. 7.

References

Antoniadis P, Courcoubetis C, Mason R (2004) Comparing economic incentives in peer-to-peer networks. Comput Netw 46:133–146

Cao X-R, Shen H-X, Milito R, Wirth P (2002) Internet pricing with a game theoretical approach: concepts and examples. IEEE/ACM Trans Netw 10:208–216

Chang C-S (1998) On deterministic traffic regulation and service guarantees: a systematic approach by filtering. IEEE Trans Inf Theory 44:1097–1110

Chang C-S, Cruz RL, Le Boudec J-Y, Thiran P (2002) A min, + system theory for constraint traffic regulation and dynamic service guarantees. IEEE/ACM Trans Netw 10(6):805–817

Cisco (2006) Configuring quality of service for voice. Available at: http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fvvfax_c/vvfqos.htm

Cerruti S, Wright C (2002) ISP bandwidth billing—how to make more or pay less. Rating matters, vol 13. Available at: http://www.servicelevel.net/rating_matters/matterslist.htm

CFDynamics (2007) available at: http://www.cfdynamics.com/cfdynamics/dedicated/dedicatedhosting basic.cfm

Courcoubetis C, Weber R (2003) Pricing communication networks: economics, technology and modelling. Wiley, New York

Fukuda K, Amaral LAN, Stanley HE (2003) Dynamics of temporal correlation in daily Internet traffic. In: Proceedings of the GLOBECOM, pp 4069–4073

Fulp EW, Reeves DS (2004) Bandwidth provisioning and pricing for the networks with multiple classes of service. Comput Netw 46:41–52

Goldenberg DK, Qiu L, Xie H, Yang YR, Zhang Y (2004) Optimizing cost and performance for multihoming. In: Proceedings of the SIGCOMM, pp 79–92

Keon NJ, Anandalingam G (2003) Optimal pricing for multiple services in telecommunications networks offering quality-of-service guarantees. IEEE/ACM Trans Netw 11:66–80

Li T, Iraqi Y, Boutaba R (2004) Pricing and admission control for QoS-enabled Internet. Comput Netw 46:87–110

Ma M, Hamdi M (2000) Providing deterministic quality-of-service guarantees on WDM optical networks. IEEE J Sel Areas Commun 18:2072–2083

Net1plus.com (1995) available at: http://www.net1plus.com

NLANR (2005) NLANR PMA, special traces archive. Available at: http://pma.nlanr.net/Special/

Odlyzko AM (2004) The evolution of price discrimination in transportation and its implications for the Internet. Rev Netw Econ 3:323–346

Ros D, Tuffin B (2004) A mathematical model of the Paris metro pricing scheme for charging packet networks. Comput Netw 46:73–85

Salehi JD, Zhang S-L, Kurose J, Towsley D (1998) Supporting stored video: reducing rate variability and end-to-end resource requirements through optimal smoothing. IEEE/ACM Trans Netw 6(4):397–410

Service Level Corporation (2002) http://www.servicelevel.net/rating_matters/matterslist.htm

Shakkottai S, Srikant R (2005) Economics of network pricing with multiple ISPs. In: Proceedings of the INFOCOM, pp 184–194

The Internet NG Project (2002) available at: http://ing.ctit.utwente.nl/

Wang H, Xie H, Qiu L, Silberschatz A, Yang YR (2005) Optimal ISP subscription for Internet multihoming: algorithm design and implication analysis. In: Proceedings of the INFOCOM, pp 2360–2371

Wang X, Schulzrinne H (2000) An integrated resource negotiation, pricing, and QoS adaptation framework for multimedia applications. IEEE J Sel Areas Commun 18:2514–2529

Yaiche H, Mazumdar RR, Rosenberg C (2000) A game theoretic framework for bandwidth allocation and pricing in broadband networks. IEEE/ACM Trans Netw 8:667–678