Optimizing Network Bandwidth Costs on the Internet

Micah Adler Ramesh Sitaraman Harish Venkataramani
Department of Computer Science,
University of Massachusetts Amherst,
Amherst, MA 01003, USA.
{micah,ramesh,harish}@cs.umass.edu

September 28, 2004

Abstract

Enterprises and Content Delivery Networks (CDNs) contract with multiple Internet Service Providers (ISPs) to ensure fault-tolerant access to the Internet. The total bandwidth cost incurred across the multiple network contracts is a sizable portion of the variable costs incurred by an enterprise or CDN for delivering internet services in a reliable fashion. This paper initiates a new area of research in internet traffic management that promises to be both algorithmically rich and practically important. We consider routing internet traffic via ISPs, where each ISP charges on the basis of the average, the maximum, or the 95^{th} percentile of the traffic routed through its uplink during the billing period. First, we devise an optimal offline algorithm that routes traffic to achieve the minimum total cost, when the network contracts charge either on a maximum or an average basis. Further, we devise a deterministic (resp., randomized) online algorithm that achieves cost that is within a factor of 2 (resp., $\frac{e}{e-1}$) of the optimal offline cost. The competitive ratios achieved are the best possible for both the deterministic and the randomized case. Finally, we show that including network contracts that charge based on the the 95^{th} percentile of the traffic renders finding the optimal solution NP-hard, even in the offline case.

1 Introduction

Over the past years, the internet has emerged as the business-critical medium for communication. Enterprises today rely partially or even entirely on the internet to communicate with their vendors and clients. A media company such as CNN or an e-commerce portal such as Amazon reach their world-wide clients over the public internet. And, increasingly even traditional brick-and-mortar enterprises transact with their vendors and clients using extranet B2B portals. For many enterprises today, even a 10-minute downtime or poor performance of its internet services during a peak period could mean millions of dollars of lost revenue! As billions of dollars are transacted on the internet every day, there is a sharp focus on technology that can deliver a better performing and more available internet at the least cost.

The internet is a network of networks, where each network is managed by an Internet Service Provider (ISP) who builds and manages the routers, links, and other networking infrastructure. As such, there are tens of thousands of ISPs that constitute the internet today, ranging from large Tier-1 providers with a global presence (such as Level 3, ATT, Sprint, WorldCom), national providers (such as China Telecom, VSNL in India, SingTel in Singapore), regional providers (such as Earthnet), and local ISPs. Entities that wish to access the internet buy internet connectivity from one or more ISPs. Network connectivity is bought and sold in the common currency of *bandwidth*, usually measured in Megabit-per-second (Mbps), under agreements that are known as *network contracts.*

An enterprise requiring high-levels of availability and performance for their internet services face a fundamental challenge. The internet itself was designed as a best-effort delivery network with no guarantees on uptime, quality of service, or performance. And, an enterprise that relies on just one ISP for connectivity to the internet runs the risk of that ISP becoming the single point of failure causing an outage. Therefore, it has recently become commonplace for large enterprises to employ a technique called *multihoming* [2], where the enterprise contracts with multiple ISPs to provide redundant internet access for its origin infrastructure, including its web servers, application servers, and back-end databases. The enterprise would then route traffic to and from its origin via the multiple uplinks, so as to minimize bandwidth costs and maximize availability and performance. Recently, commercial offerings from RouteScience [3] and Internap [4] offer products that help the enterprise optimize traffic across the different ISPs that provide network connectivity.

A complimentary and a more comprehensive approach for achieving a reliable internet service is for the enterprise to contract with a Content Delivery Network (CDN) to host their web site. Examples of such CDNs include Akamai [1, 6] and Speedera [5]. A CDN is a large-scale distributed system with servers hosted in potentially thousands of ISPs. A CDN negotiates network contracts to buy bandwidth from each of those ISPs. An end-user accessing content hosted on the CDN is directed to an appropriate server at one of the contracted ISPs that can serve the content. Specifically, traffic from end-users accessing content in a CDN is directed to servers in the different ISPs, so as to optimize availability and performance for the end-user and minimize bandwidth costs for the CDN.

The primary focus of this paper is building an algorithmic framework for the all-too-important problem of routing traffic to minimize overall bandwidth costs across multiple network contracts. The model and algorithms proposed here find application in the multihoming context, where an enterprise would like to minimize the total cost incurred in its contracts with ISPs. But, this work is even more applicable in the context of a CDN that wishes to optimize the overall total cost incurred across its numerous network contracts. As such, bandwidth costs incurred in the network contracts with the ISPs account for a significant portion of the variable costs (i.e., Cost of Goods Sold) for operating the CDN. Therefore, managing the traffic to reduce bandwidth costs is critically important.

1.1 Network Contracts

A first important step in our study is accurately modeling a network contract with an ISP. While a network contract is a complex legal document, there are three important parameters that provide a simple yet realistic model for designing applicable optimization algorithms.

Type The contract type dictates how the ISP will bill for the traffic that is sent over its link.
The billing period (typically a month) is divided into M 5-minute time buckets (typically,

$M = 8640$), and the total traffic sent on the link is averaged within each 5-minute bucket. The three types of contracts we study are AVG, MAX, and 95th contracts where the billable traffic is computed as the average, maximum or the 95th percentile respectively of the 5-minute-bucket-averages in the billing period. The AVG and 95^{th} contracts are the industry standards accounting for most network contracts in existence today. Routing traffic demand on the internet is imprecise, since the offered load is often hard to estimate and the controls are imprecise (for instance, when web traffic is moved from one ISP to another ISP, it may take several minutes for the move to take effect depending on DNS TTLs and browser behavior). Due to the imprecision in both traffic estimation and control, 95^{th} contracts are handled as though they were MAX contracts in practice. Hence, the great importance of studying MAX contracts. Further, as we will see MAX contracts are more tractable and provide good insights into the underlying optimization.

Unit Cost *Unit cost* is the cost per Mbps of billable traffic. Let $x_1 \geq x_2 \geq \cdots \geq x_M$ be the average traffic within each of the M 5-minute buckets during the billing period, placed in descending order. For an AVG contract, the bill for the month is $C_{AVG} * (\sum_i x_i)/M$, where C_{AVG} is the unit cost. For a MAX contract, the bill for the month is $C_{MAX} * x_1$, where C_{MAX} is the unit cost. Likewise, for a 95^{th} contract, the bill for the month is $C_{95th} * x_{\frac{M}{20}}$.

Capacity. The capacity P is the maximum bandwidth (in Mbps) that one can send through the uplink of the ISP.

In addition to these three parameters, an additional parameter called the Committed Information Rate (CIR) is important to model. CIR represents the committed amount of billable traffic that must be sent through an ISP. The CIR is paid for in advance, whether or not it is used. While the results presented in this paper do not consider CIR, accounting for CIRs is an important direction for future work.

1.2 Prior Work

There is no work that the authors are aware of that model network contracts and study the problem of minimizing bandwidth costs incurred in these contracts. However, considering the practical importance of the problem in recent years, heuristic implementations exist for similar problems in a real-world setting. The techniques used here are rooted in the extensive literature on online algorithms [13, 14]. Specifically, the decision of whether to use a MAX versus an AVG contract is similar to a generalization of the buy-versus-rent decision in the classical Ski Rental Problem [12, 11].

1.3 Our Contributions

The first contribution of the paper is the modeling and formulation of an area of great practical importance with a rich potential for algorithmic investigation. In Section 2, we derive an optimal offline algorithm that routes traffic to a set of ISPs with AVG and MAX contracts such that the total cost is minimized. The offline algorithm assumes that the traffic that needs to be routed for the entire billing period is known in advance. While this is not an assumption that holds in practice, the offline optimal algorithm places a lower bound on the cost that is achievable by any online algorithm that is used in practice. Among other things, this is valuable in bounding the amount of cost reduction possible by investing in better online algorithms for traffic management. In Section 3, we turn to online algorithms that know only the current and the past traffic levels, and is unaware of any events in the future. Specifically, we devise a deterministic online algorithm that is at most a factor of 2 in cost from the optimal offline solution. Further, in Section 4, we devise a randomized online algorithm that has an expected cost that is a factor of $\frac{e}{e-1}$ from optimal. In both cases, we show that the competitive ratios are the best possible. Finally, we show that optimizing costs for 95^{th} percentile contracts is NP-hard, which differentiates it from the MAX and AVG contracts.

1.4 Problem Description

The internet traffic management problem is modeled as follows. The billing period (typically one month) is divided into M 5-minute time buckets. We model the incoming traffic as a sequence b_t, $1 \leq t \leq M$, where b_t is the average traffic (Mbps) in the time bucket t. Each b_t represents the average traffic demand from end-users that must served from the contracted ISPs at that time bucket. At any time t, a traffic routing algorithm partitions the incoming traffic b_t and assigns y_t^j Mbps to ISP_j such that $\sum_j y_t^j = b_t$. Further, it ensures that capacity constraints are met at each ISP_j and at each time $1 \leq t \leq M$, i.e., $y_t^j \leq P_j$, where P_j is the capacity of ISP_j.

An offline algorithm knows the entire time-ordered input sequence of traffic demands, $I = \langle b_t \rangle$, $1 \leq t \leq M$, for the entire billing period. And, it makes traffic routing decisions based on this complete knowledge. An online algorithm on the other hand makes routing decisions at time t knowing only b_j, $1 \leq j \leq t$, i.e., knowing only the past and current values.

In this paper, we study both offline and online algorithms for traffic management that optimize the total cost incurred in the network contracts for the billing period. We use the notion of competitive ratio to bound the cost $C_A(I)$ of an online algorithm A in terms of the optimal offline cost of $C_{OPT}(I)$. In particular, a deterministic online algorithm A is said to be c-competitive if there exists a constant α such that for all input sequences I, $C_A(I) \leq c \cdot C_{OPT}(I) + \alpha$. A similar competitive notion applies to randomized online algorithms where the *expected* value of the cost is used instead.

2 The Offline Algorithm

In this section, we derive an optimal offline algorithm that routes traffic to ISPs with AVG or MAX contracts with minimum cost. Assume that we are given contracts from m MAX ISPs Max_i, $1 \leq i \leq m$, such that $C_{Max_1} < C_{Max_2} < \cdots < C_{Max_m}$. Further, assume that we are given contracts from n AVG ISPs Avg_i, $1 \leq i \leq n$, such that $C_{Avg_1} < C_{Avg_2} < \cdots < C_{Avg_n}$. Note that without loss of generality we may assume that the costs of two MAX (resp., AVG) ISPs are not equal, since one can merge two contracts of the same cost and type to form one contract with the sum of the capacities. We start by proving a series of lemmas that characterize an optimal solution. We will then use this characterization to efficiently compute the optimal solution.

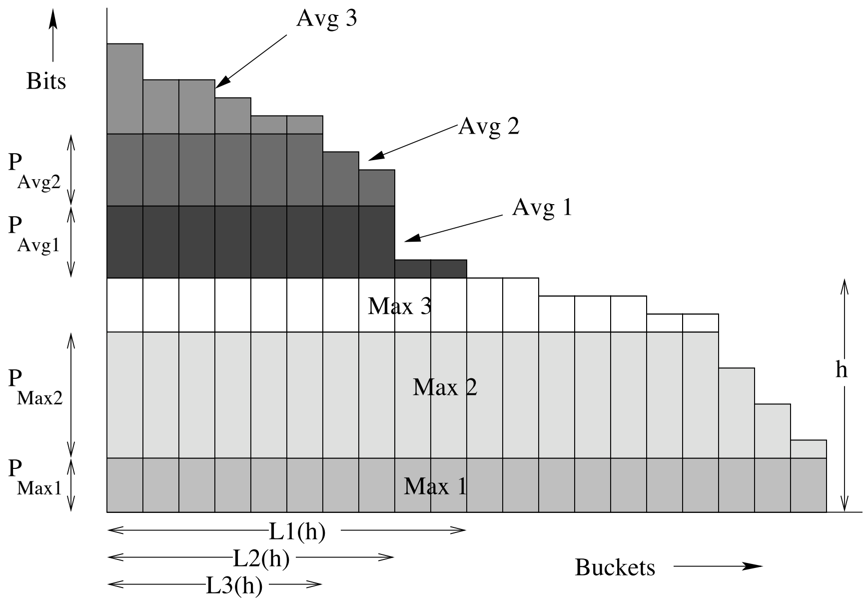

Figure 1: The structure of an optimal solution

Lemma 1 *In any optimal solution, ISP Avg_i is not used in a time bucket unless each ISP Avg_j, $j < i$, is used to its full capacity.*

Proof: Assume there is an optimal solution contrary to this lemma. Moving traffic from Avg_i to a cheaper ISP Avg_j that has residual capacity left reduces the cost. Contradiction. $\square$

Define the threshold t_{Max_i} of an ISP Max_i to be the maximum traffic routed during the billing period through that ISP.

Lemma 2 *In any optimal solution, threshold $t_{Max_i} > 0$ only if $t_{Max_j} = P_{Max_j}$ for all $j < i$, where P_{Max_j} is the capacity of the ISP Max_j.*

Proof: Assume there exists an optimal solution contrary to this lemma. Let Max_j, $j < i$, be an ISP such that $t_{Max_j} < P_{Max_j}$. We can now move traffic of up to $P_{Max_j} - t_{Max_j}$ in every time bucket from ISP Max_i to the cheaper ISP Max_j. This results in a reduction of the threshold of Max_i, and hence a reduction in total cost. Contradiction. $\square$

Lemma 3 *There exists an optimal solution in which Max_i is not used in a time bucket until each ISP Max_j, $j < i$, has been used to its full capacity of P_{Max_j}.*

Proof: Suppose that the lemma does not hold for an optimal solution in some time bucket t. We show how to reroute the traffic in that time bucket to create a new optimal solution with same cost that obeys the lemma in that time bucket. Let i be the largest value such that Max_i is used in time bucket t. Using Lemma 2 and the fact that $t_{Max_i} > 0$, it follows that $t_{Max_j} = P_{Max_j}$, for all $j < i$. Therefore, one can reroute the traffic in time bucket t by filling the ISPs to capacity in sequential order starting from Max_1. This does not increase any of the thresholds and hence does not affect the overall cost. Thus, the new solution after the rerouting is also optimal. □

Lemma 4 *In the optimal solution, in any time bucket an AVG ISP is used only if all MAX ISPs are used to their respective thresholds for the billing period.*

Proof: Assume to the contrary. If there exists an optimal solution where ISP Avg_i receives $x > 0$ units of traffic in a time bucket, but some ISP Max_j is used less than its threshold t_{Max_j} by $y > 0$ units. By moving $\min\{x, y\} > 0$ units of traffic from Avg_i to Max_j, the total cost of ISP Avg_i decreases while the cost of Max_j remains the same. Thus, the overall cost decreases, which leads to a contradiction. □

As a consequence of the above lemmas, we have shown that there exists an optimal solution that is structured as in Figure 1. The sequence of vertical bars in the figure represent traffic a_i in time bucket i sorted in non-increasing order, i.e., $a_1 \geq a_2 \geq \cdots \geq a_M$. As shown in the figure *max threshold* h is the sum of the thresholds of all the MAX contracts. Define max threshold h_{OPT} to be equal to the value of h in the optimal solution. Note that in each bucket, MAX ISPs are filled to capacity in the decreasing order of cost until that height h_{OPT} is reached or there is no more traffic to be served. After height h_{OPT} is reached, AVG ISPs are filled to capacity in the decreasing order of cost until there is no remaining traffic to be served.

To compute the optimal it suffices to compute the optimal value of Max Threshold h that minimizes total cost.

Lemma 5 *Let $Dec(h)$ (resp., $Inc(h)$) represent the decrease (resp., increase) in the total cost of the AVG (resp. MAX) contracts when the Max Threshold is increased from h to $h + 1$. $Dec(h)$ (resp., $Inc(h)$), is a non-increasing (resp., non-decreasing) function of h.*

Proof: Let $L_i(h)$ be the number of buckets in which traffic is sent through the AVG ISP Avg_i when the Max Threshold is h. $Dec(h)$ to be the decrease in the total cost incurred in AVG ISPs if the Max Threshold is changed h to $h + 1$. Now $Dec(h) = C_{Avg_1}(L_1(h) - L_2(h)) + C_{Avg_2}(L_2(h) - L_3(h)) + \cdots + C_{Avg_{n-1}}(L_{n-1}(h) - L_n(h)) + C_{Avg_n} L_n(h)$. $L_i(h)$ is a monotonically non-increasing function of h as we are sending more through the MAX ISPs. In the following equations as the coefficients of $L_i(h)$ are positive constants we can say that $Dec(h)$ is also a non-increasing function of h.

$$\begin{aligned} Dec(h) &= C_{Avg_1}(L_1(h) - L_2(h)) + \cdots + C_{Avg_{n-1}}(L_{n-1}(h) - L_n(h)) + C_{Avg_n} L_n(h) \\ &= L_1(h)(C_{Avg_1}) + L_2(h)(C_{Avg_2} - C_{Avg_1}) + \cdots + L_n(h)(C_{Avg_n} - C_{Avg_{n-1}}) \end{aligned}$$

Note that $Inc(h)$ is the cost of the MAX ISP that is used for routing the unit of traffic at height $h + 1$. Since the MAX ISPs are used in the increasing order of cost, $Inc(h)$ is a non-decreasing function of h. □.

Theorem 6 *The offline optimal solution can be computed in $O((n+m)M \log(nmM))$ time, where m is the number of MAX ISPs, n is the number of AVG ISPs and M is the total number of buckets in the billing period.*

Proof: Note that a value of h is permissible if and only if it is feasible to route all traffic at or below height h using MAX contracts, and all traffic above h using AVG contracts. Note that as long as $Dec(h) > Inc(h)$, we can decrease the total cost by incrementing h. From Lemma 5, it follows that the optimal value of $h = h_{OPT}$ is either the lowest permissible value of h such that $Dec(h) \leq Inc(h)$ or if no such h exists then it is the largest permissible value of h.

By the definition of h_{OPT}, $Dec(h) > Inc(h)$ for all permissible $h < h_{OPT}$ and by Lemma 5 for all permissible $h \geq h_{OPT}$ we have $Dec(h) \leq Dec(h_{OPT}) \leq Inc(h_{OPT}) \leq Inc(h)$. Thus one can find out an optimal value, h_{OPT}, by doing binary search on just those values of h where the values where $Dec(h)$ or $Inc(h)$ differ from $Dec(h-1)$ and $Inc(h-1)$ respectively.

$Inc(h)$ differs from $Inc(h-1)$ whenever $h = \sum_{j=1}^{k-1} P_{Max_j}$ for some k as now a new MAX ISP is used. A sorted array of size m containing such values of h can be calculated in $O(m)$ time. The value of $Dec(h)$ changes only if for some i the value of $L_i(h)$ changes on increasing the Max Threshold from $h-1$ to h. $L_i(h)$ changes only if $h + \sum_{j=1}^{i-1} P_{Avg_j} = b_k$ for some k. For each i a sorted array of size at most M such values of such h can be calculated in $O(M)$ time using a pre-calculated array of $\sum_{j=1}^{i-1} P_{Avg_j}$ for $i = 1, 2, \ldots, n$. This array can be calculated in $O(n)$ time beforehand. The n sorted arrays of size at most M, where $Dec(h)$ can change, and the sorted array of size at most m, where $Inc(h)$ can change, can be merged into a single sorted array of size at most $nM + m$ in $O(nM + m)$ time.

The binary search on this restricted set takes $\log(nM + m)$ iterations. In each iteration we calculate on the fly the values of $Inc(h)$ and $Dec(h)$ for the new value of h. $Inc(h) = C_{Max_i}$ if $\sum_{j=1}^{i-1} P_{Max_j} \leq h < \sum_{j=1}^{i} P_{Max_j}$ Thus $Inc(h)$ can be calculated in $O(\log m)$ time searching for h using binary search on a pre-calculated array of size m where the i^{th} element is the sum of the capacities of the i cheapest MAX ISPs. $L_k(h)$ can be found in $O(\log M)$ time by searching for $h + \sum_{j=1}^{k-1} P_{Avg_j}$ in the array $b_1, b_2, \ldots, b_M$ using binary search and returning the index i such that $b_i \leq h + \sum_{j=1}^{k-1} P_{Avg_j} < b_{i-1}$. The array containing $\sum_{j=1}^{k-1} P_{Avg_j}$ is calculated only once and used repeatedly. Thus $Dec(h)$ can be calculated in $O(n \log M)$ time. Thus the time required per iteration is $O(n \log M + \log m)$. Thus the total time required for finding the optimal value of h is $O((n \log M + \log m)\log(nM+m) + nM + m + n + m\log m + n \log n + M \log M) = O((n \log M + \log m)\log(nM + m) + nM + m\log m + M \log M)$. The additional time of $O(m \log m + n \log n + M \log M)$ is used is for the initial sorting of the traffic buckets and costs. Once h_{OPT} has been found the optimal traffic assignment for each bucket can be output in $O(M(n+m))$ time.□

3 A Deterministic Online Algorithm

In this section, we present a 2-competitive deterministic online algorithm A that routes traffic to AVG and MAX ISPs. The algorithm is given a time-ordered sequence of traffic demands, $I = \langle b_1, b_2, \cdots, b_{M-1}, b_M \rangle$. At a given time bucket t, algorithm A does the following:

1. Run the offline algorithm OPT described in Section 2 on the input $\langle b_1, b_2, \cdots, b_t, 0, 0, \cdots, 0 \rangle$. That is, run the offline on a prefix of the input assuming all future time buckets have zero traffic.

2. Route the current traffic b_t in same manner as OPT.

First, we show that the Max Threshold, i.e., the sum of the thresholds incurred in the MAX contracts, can only increase with time as we progress through the month.

Lemma 7 *Let h_i be the Max Threshold of OPT on input $\langle b_1, b_2, \cdots, b_i, 0, 0, \cdots 0\rangle$. Then, for all $1 \le t \le M-1$, $h_t \le h_{t+1}$.*

Proof: Let $Inc_i(h)$ and $Dec_i(h)$ be defined as in Section 2 for input $\langle b_1, b_2, \cdots, b_i, 0, 0, \cdots 0\rangle$. By definition, h_t is either the lowest permissible value of h such that $Dec_t(h) \le Inc_t(h)$ or if no such h exists then it is the largest permissible value of h for input sequence $\langle b_1, b_2, \cdots, b_t, 0, 0, \cdots 0\rangle$. Note that

$$Inc_{t+1}(h_t) = Inc_t(h_t) \tag{1}$$

Further, note that

$$Dec_{t+1}(h_t) \ge Dec_t(h_t), \tag{2}$$

since the two are equal if $b_{t+1} \le h_t$ and the LHS is greater than the RHS otherwise. Observing that h_t is also a permissible value for the input sequence $\langle b_1, b_2, \cdots, b_t, b_{t+1}, 0, 0, \cdots 0\rangle$, equations 1 and 2 imply that h_{t+1} either equals h_t or is larger than it. $\square$

Theorem 8 *The competitive ratio of the deterministic online algorithm A is 2.*

Proof: The total cost C_A of algorithm A equals the sum of the cost $C_{A,Avg}$ incurred in the AVG contracts and the cost $C_{A,Max}$ incurred in the MAX contracts. Note that the final threshold h_M of A equals the threshold h_{OPT} computed by the offline optimal algorithm OPT. Therefore,

$$C_{A,Max} = C_{OPT,Max} \le C_{OPT} \tag{3}$$

Let $C^t_{A,Avg}$ be the cost incurred in AVG ISPs by algorithm A during the first t time buckets. Similarly, let C^t_{OPT} be the total cost incurred by the optimal offline algorithm OPT when provided an input of $\langle b_1, b_2, \cdots, b_t, 0, 0, \cdots, 0\rangle$. We prove by induction on t, that $C^t_{A,Avg} \le C^t_{OPT}$.
Base Case When $t = 1$, algorithm A runs OPT on the first input and behaves identical to it. Therefore,

$$C^1_{A,Avg} = C^1_{OPT,Avg} \le C^1_{OPT}$$

Inductive Case: Assume that the hypothesis is true till t. So $C^t_{A,Avg} \le C^t_{OPT}$. As C^t_{OPT} is the optimal offline solution for the input $\langle b_1, b_2, \cdots, b_t, 0, 0, \cdots, 0\rangle$, $C^t_{A,Avg} \le C^t_{OPT} \le$ the cost of the optimal offline solution with Max Threshold as h_{t+1} for the same input. The contribution in the cost of $C^{t+1}_{A,Avg}$ and C^{t+1}_{OPT} of sending part of the data in the $t+1^{th}$ interval through the AVG ISPs is the same. This is because in both cases only the data more than h_{t+1} is sent through the AVG ISPs. Adding this cost to the extremities of the inequality given above we get $C^{t+1}_{A,Avg} \le C^{t+1}_{OPT}$. This completes the induction. Therefore,

$$C_{A,Avg} = C^M_{A,Avg} \le C^M_{OPT} = C_{OPT} \tag{4}$$

Thus, combining equations 3 and 4, $C_A = C_{A,Max} + C_{A,Avg} \le 2C_{OPT}$ $\square$

Theorem 9 *The competitive ratio of 2 achieved by Algorithm A is the best possible for any deterministic online algorithm.*

Proof: We prove a lower bound by showing that the Ski Rental problem[11] is a special case of the traffic routing problem. Given a ski rental problem where the cost of renting a ski is 1 and the cost of buying a ski is p, the optimal strategy when you ski k times is to buy skis in the beginning if $k \geq p$, and rent otherwise. Given an instance of the ski rental problem we create an instance of the traffic routing problem with one MAX ISP of cost p and one AVG ISP of cost M, where $M \geq k$ is the number of buckets in the billing period. The input traffic $b_t = 1$, if $1 \leq t \leq k$, and zero for $k < t \leq M$. The capacity of each ISP is 1. It is easy to verify that in the original ski rental problem the optimal solution is to buy skis if and only if the optimal solution for bandwidth cost minimization problem is to use MAX ISP to send the entire data. Similarly renting skis is optimal if and only if AVG ISP is used to send the entire data.

If for any $\epsilon > 0$ if there exists a deterministic online algorithm with competitive ratio of $2 - \epsilon$ we can use it to get a $2 - \epsilon$ competitive deterministic online algorithm for the ski rental problem using the construction given above. This contradicts the fact that ski rental problem has a lower bound on the competitive ratio of a deterministic online algorithm of $1 + \frac{\lceil p \rceil - 1}{p}$ which $\longrightarrow 2$ as $p \longrightarrow \infty$[11, 12]. $\square$

4 A Randomized Online Algorithm

In this section we describe an $e/(e-1)$ competitive randomized online algorithm A_{Rand} which

1. Picks z between 0 and 1 according to the probability density function $p(z) = \frac{e^z}{e-1}$.
2. Routes the traffic using the deterministic online algorithm A_z.

The online algorithm A_z is given a time-ordered sequence of traffic demands, $I = \langle b_1, b_2, \cdots, b_{M-1}, b_M \rangle$. At a given time bucket t, algorithm A_z does the following:

1. Run the offline algorithm $OPT(z)$ described in Section 2 on the input $\langle b_1, b_2, \cdots, b_t, 0, 0, \cdots, 0 \rangle$ but with the costs of all MAX ISPs multiplied by z.
2. Route the current traffic b_t in same manner as $OPT(z)$.

Define $C_{OPT}(z)$ to be the cost of the optimal offline solution with the same input but with the costs of all MAX ISPs multiplied by z. Let $C_{OPT,Avg}(z)$(resp., $C_{OPT,Max}(z)$) be the contribution in $C_{OPT}(z)$ due to the AVG (resp., MAX) ISPs. Similarly define $C_{A_z,Avg}$(resp., $C_{A_z,Max}$) to be the contribution in C_{A_z}, the total cost due to algorithm A_z, due to the AVG (resp., MAX) ISPs. Note that C_{A_z} and $C_{A_z,Max}$ are charged by the actual cost of the MAX ISPs but $C_{OPT}(z)$ and $C_{OPT,Max}(z)$ have a discounting factor of z. Also A_1 is the deterministic online algorithm A given in section 3.

Lemma 10

$$C_{A_z,Avg} \leq C_{OPT}(z) \tag{5}$$

Proof: This proof is similar to the proof given for equation 4, $C_{A,Avg} \leq C_{OPT}$, given in theorem 8. $\Box$

Lemma 11

$$zC_{A_z,Max} = C_{OPT,Max}(z) \tag{6}$$

Proof: Again this can be proved in the same way as equation 3, $C_{A,Max} = C_{OPT,Max}$, was proved in theorem 8. The only difference is that in $C_{OPT}(z)$ the costs of the MAX ISPs are multiplied by z and in A_z they are not. $\Box$

Lemma 12 *For* $0 \leq z \leq 1$,

$$C_{OPT}(1) - C_{OPT}(z) \geq \int_z^1 C_{A_w,Max} dw$$

Proof: For any v such that $0 \leq z \leq v \leq 1$,

$$\begin{aligned} C_{OPT}(v) &= C_{OPT,Max}(v) + C_{OPT,Avg}(v) \\ &= vC_{A_v,Max} + C_{OPT,Avg}(v) \\ d(C_{OPT}(v)) &= dv \cdot C_{A_v,Max} + v \cdot d(C_{A_v,Max}) + d(C_{OPT,Avg}(v)) \end{aligned} \tag{7}$$

Define $h(w)$ to be the Max Threshold in the optimal offline solution ($C_{OPT}(w)$) when the cost of all MAX ISPs are multiplied by w. $h(w)$ is a non-increasing function of w. Also let C_{Max_w} be the original cost of the MAX ISP with highest cost that was used in optimal offline solution $C_{OPT}(w)$ (or in C_{A_w}). As the actual cost of any MAX ISP used in the gap between $h(v)$ and $h(v+dv)$ would be at most C_{Max_v}

$$-d(C_{A_v,Max}) = C_{A_v,Max} - C_{A_{v+dv},Max} \leq C_{Max_v} \cdot (h(v) - h(v+dv)) \tag{8}$$

The actual cost of any MAX ISP used in the gap between $h(v)$ and $h(v+dv)$ is at least $C_{Max_{v+dv}}$. Thus in the optimal solution when the cost of the MAX ISPs have been multiplied by v decreasing the Max Threshold from $h(v)$ by $h(v) - h(v+dv)$ decreases the cost due to the MAX ISPs by at least $vC_{OPT}(v+dv) * (h(v) - h(v+dv))$. The corresponding increase in the cost due to the AVG ISPs should be at least this much otherwise we contradict that $C_{OPT}(v)$ is optimal cost. Thus

$$d(C_{OPT,Avg}(v)) = C_{OPT,Avg}(v+dv) - C_{OPT,Avg}(v) \geq vC_{Max_{v+dv}} \cdot (h(v) - h(v+dv)) \tag{9}$$

Substituting equations 8,9 in equation 7

$$\begin{aligned} d(C_{OPT}(v)) &\geq dv \cdot C_{A_v,Max} - v \cdot (C_{Max_{v+dv}} - C_{Max_v}) \cdot (h(v+dv) - h(v)) \\ &= dv \cdot C_{A_v,Max} - v \cdot d(C_{Max_v}) \cdot d(h(v)) \end{aligned}$$

Integrating v from z to 1 and using the fact that $C_{OPT}(v)$ is a continuous function and that the integral of the product of two differentials is 0, we get $C_{OPT}(1) - C_{OPT}(z) \geq \int_z^1 C_{A_v,Max} dv$. $\Box$

Corollary 13

$$C_{OPT}(1) \quad \geq \quad \int_0^1 C_{A_w,Max} dw$$

Theorem 14 *The competitive ratio of the randomized online algorithm A_{Rand} is $e/(e-1)$*

Proof: Define $P(z) = \int_0^z p(w)dw$. Then

$$\begin{aligned}
C_{A_z} &= C_{A_z,Max} + C_{A_z,Avg} \\
&\leq C_{A_z,Max} + C_{OPT}(1) - \int_z^1 C_{A_w,Max} dw \\
&\quad \text{by lemma 10 and lemma 12} \\
E[C_{A_{Rand}}] &= \int_0^1 C_{A_z} p(z) dz \\
&\leq C_{OPT}(1) + \int_0^1 C_{A_z,Max} p(z) dz - \int_0^1 p(z) (\int_z^1 C_{A_w,Max} dw) dz \\
&= C_{OPT} + \int_0^1 C_{A_z,Max} p(z) dz - \int_0^1 C_{A_w,Max} (\int_0^w p(z) dz) dw \\
&\quad \text{by changing the order of integration} \\
&= C_{OPT} + \int_0^1 (p(z) - P(z)) C_{A_z,Max} dz \\
\frac{E[C_{A_{Rand}}]}{C_{OPT}} &\leq 1 + \frac{\int_0^1 (p(z) - P(z)) C_{A_z,Max} dz}{\int_0^1 C_{A_z,Max} dz}
\end{aligned}$$

By corollary 13. Setting $p(z) = \frac{e^z}{e-1}$ and $P(z) = \frac{e^z - 1}{e-1}$ in the above result we prove the theorem. □

Theorem 15 *The competitive ratio of $e/(e-1)$ achieved by Algorithm A_{Rand} is the best possible for any randomized online algorithm.*

Proof: As in theorem 9 we use the fact that this problem is a generalization of the ski rental problem. The ski rental problem has lower bound on the competitive ratio of a randomized online algorithm of $e'_p/(e'_p - 1)$ where $e'_p = (1 + \frac{1}{p-1})^p$ when p, the ratio of the cost of buying to the cost of selling, is an integer. The algorithm which achieves this is similar to the randomized online algorithm for the snoopy caching problem[12]. Also $e'_p/(e'_p - 1) < e/(e-1)$ but tends to $e/(e-1)$ as p tends to ∞.

If for any $\epsilon > 0$ if there exists a $e/(e-1) - \epsilon$ competitive randomized algorithm for this problem then by the construction in theorem 9 we get a $e/(e-1) - \epsilon$ competitive randomized algorithm for the ski rental problem. A contradiction. □

5 Hardness of the 95^{th} Percentile Contracts

Including network contracts that charge based on the the 95^{th} percentile of the traffic renders finding the optimal solution NP-hard, even in the offline case. In fact the following easier problem is also NP-Hard.

Theorem 16 *Finding whether one can route the entire traffic with zero cost in a system consisting of n* 95^{th} *percentile ISPs is NP-Complete in the strong sense.*

Proof: The proof involves a straight forward reduction from the Bin Covering Problem[7] to this problem. Which is known NP-complete in the strong sense. Details omitted.□

References

[1] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl. Globally distributed content delivery. IEEE Internet Computing, September 2002, pp. 50–58.

[2] A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman. A Measurement-Based Analysis of Multihoming. Proceedings of the 2003 ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM), August 2003.

[3] Routescience. http://www.routescience.com.

[4] Internap: Premise-based Route Optimization. http://www.internap.com/products/route-optimization.htm.

[5] Speedera. http://www.speedera.com.

[6] Akamai. http://www.akamai.com.

[7] A. B. Assmann, D. J. Johnson, D. J. Kleitman, and J. Y.-T. Leung. On a dual version of the one-dimensional bin packing problem. *Journal of Algorithms*, 5(4):502–525, 1984.

[8] D. R. Dooly, S. A. Goldman, and S. D. Scott. TCP dynamic acknowledgment delay: Theory and practice. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC-98)*, pages 389–398, New York, 23–26 1998. ACM Press.

[9] R. Fleischer. On the Bahncard problem. *Theoretical Computer Science*, 268(1):161–174, 2001.

[10] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[11] A. R. Karlin, C. Kenyon, and D. Randall. Dynamic TCP acknowledgment and other stories about e/(e-1). In *ACM Symposium on Theory of Computing*, pages 502–509, 2001.

[12] A. R. Karlin, M. S. Manasse, L. Rudolph, and D. Sleator. Competitive snoopy caching. *Algorithmica*, 3(1):70–119, 1988.

[13] D. D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Commun. ACM*, 28(2):202–208, 1985.

[14] A. Borodin and R. El-Yaniv. Online Computation and Competitive Analysis. *Cambridge University Press*, 1998.