2011

Algorithms for Optimizing the Bandwidth Cost of Content Delivery

Micah Adler
Ramesh Sitaraman, *University of Massachusetts - Amherst*
Harish Venkataramani

Algorithms for Optimizing the Bandwidth Cost of Content Delivery

Micah Adler[c], Ramesh K. Sitaraman[*,a], Harish Venkataramani[b]

[a]*Department of Computer Science, University of Massachusetts, Amherst, MA 01003, USA*
[b]*Google Inc, 1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA*
[c]*Fluent Mobile, Ten Post Office Square, 8th Floor, Boston MA 02109, USA*

Abstract

Content Delivery Networks (CDNs) deliver web content to end-users from a large distributed platform of web servers hosted in data centers belonging to thousands of Internet Service Providers (ISPs) around the world. The bandwidth cost incurred by a CDN is the sum of the amounts it pays each ISP for routing traffic from its servers located in that ISP out to end-users. A large enterprise may also contract with multiple ISPs to provide redundant Internet access for its origin infrastructure using technologies such as multihoming and mirroring, thereby incurring a significant bandwidth cost across multiple ISPs. This paper initiates the formal *algorithmic* study of bandwidth cost minimization in the context of a large enterprise or a CDN, a problem area that is both algorithmically rich and practically very important. First, we model different types of contracts that are used in practice by ISPs to charge for bandwidth usage, including average, maximum, and 95^{th}-percentile contracts. Then, we devise an optimal offline algorithm that routes traffic to achieve the minimum bandwidth cost, when the network contracts charge either on a maximum or on an average basis. Next, 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 for maximum and average contracts. In addition, we prove that our online algorithms achieve the best possible competitive ratios in both the deterministic and the randomized cases. An interesting theoretical contribution of this work is that we show intriguing connections between the online bandwidth optimization problem and the seemingly-unrelated but well-studied ski rental problem where similar optimal competitive ratios are known to hold. Finally, we consider extensions for contracts with a committed amount of spend (known as Committed Information Rate or CIR) and contracts that charge on a 95^{th}-percentile basis.

Key words: Internet content delivery, Content delivery networks, Optimization algorithms, Bandwidth cost minimization, Network algorithms, Traffic management algorithms, Online algorithms.

1. Introduction

The Internet has emerged as a business-critical medium for enterprises to communicate with their vendors and clients. However, the Internet itself was designed as a best-effort delivery network with no guarantees on availability or performance. The Internet is a network of networks, where each network is managed independently by an Internet Service Provider (ISP) who builds and manages the routers, links, and other networking infrastructure. As such, there are more than 13,000 ISPs that constitute the Internet today, ranging from large Tier-1 providers with a global presence (such as Level 3, and ATT), national providers (such as China Telecom, and SingTel in Singapore), regional providers (such as Earthnet), and local Tier-3 ISPs. An enterprise requiring high-levels of availability for their Internet services faces a fundamental challenge. It is not sufficient for the enterprise to obtain their Internet connectivity from a single ISP, as any single ISP is prone to failure caused by router breakdowns, fiber cuts, and configuration errors. Therefore, many enterprises use strategies such as mutihoming and mirroring that allow them to access the Internet using multiple ISPs and data centers. In addition, many major enterprises use a Content Delivery Network (CDN) that is a large fault-tolerant distributed platform of web servers hosted in potentially thousands of ISPs. Examples of such CDNs include Akamai [4] and Limelight [3]. A significant fraction of the web traffic today use CDNs, including most major media, entertainment, e-commerce, and extranet portals. For a comprehensive description of the rationale for CDNs and the system architecture of Akamai's CDN, the reader is referred to [11].

1.1. CDN System Architecture

The model and results of this paper apply in several general technological contexts where cost-efficient inter-ISP traffic management is critical. But, perhaps the most important context is that of a large global CDN. We provide a brief overview of CDN architecture (see Figure 1). It is instructive to follow the actions of a typical user to see how the various system components interact to deliver content to that user.

- When the user types a URL into his/her browser, the domain name of the URL is translated by the mapping system into the IP address of an edge server to serve the content (arrow 1). To assign the user to a server, the mapping system bases its answers on large amounts of historical and current data that have been collected and processed regarding global network and server conditions, and cost. This data is used to choose an edge server that is located close to the end user.

*Corresponding author

Email addresses: micah@fluentmobile.com (Micah Adler), ramesh@cs.umass.edu (Ramesh K. Sitaraman), harishv@google.com (Harish Venkataramani)

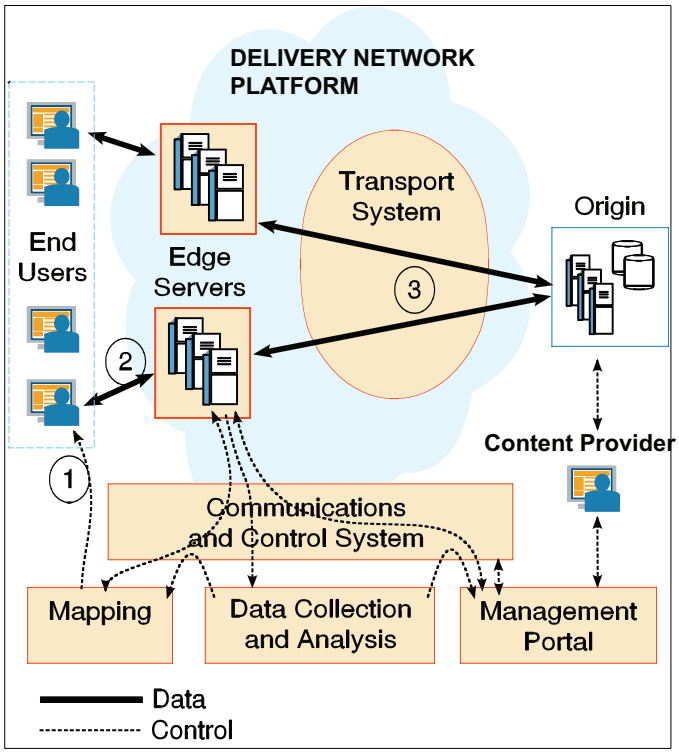

Figure 1: The system architecture of a CDN.

- Each edge server is part of the edge server platform, a large global deployment of servers located in thousands of sites around the world. These servers are responsible for processing requests from users and serving the requested content (arrow 2).

- In order to respond to a request from a user, the edge server may need to request content from an origin server (arrow 3). The transport system is used to download the required data in a reliable and efficient manner.

- The communications and control system is used for disseminating status information, control messages, and configuration updates in a fault-tolerant and timely fashion. The data collection and analysis system is responsible for collecting and processing data from various sources such as server logs, client logs, and network and server information. Finally, the management portal serves two functions. First, it provides a configuration management platform that allows an enterprise customer (i.e., Content Provider) to retain fine-grained control how their content and applications are served to the end user. In addition, the management portal provides the enterprise with visibility on how their users are interacting with their

applications and content, including reports on audience demographics and traffic metrics.

While a comprehensive description of the CDN architecture is out of the scope of the current paper (see [11] and [1] instead), we restrict our attention to one specific facet of the mapping system that optimizes the bandwidth costs incurred in routing traffic to end-users.

1.2. Bandwidth Cost Optimization

A CDN negotiates network contracts to buy Internet bandwidth from a large number of ISPs and co-locates its edge servers in those ISPs. An end-user accessing web content hosted on the CDN is directed by the CDN's mapping system to an appropriate server at one of the contracted ISPs, so as to optimize availability and performance for the end-user and to minimize bandwidth costs for the CDN. Thus, a CDN's mapping system [11] operates as an "Internet traffic cop" by controlling which portion of the traffic demand is served from which ISP. The traffic assignments happen in an online and "real-time" fashion where assignments are changed periodically at the time granularity of minutes (say, every 5 minutes).

A CDN can be viewed as a reseller of Internet bandwidth, where it pays each ISP for the traffic served from that ISP to end-users. A CDN in turn gets paid by the enterprises (i.e., content providers) for the traffic the CDN delivered on their behalf. A significant portion of the variable[1] costs of operating a CDN is the total bandwidth costs that it pays the ISPs, and minimizing this cost is the primary focus of this paper. Note that while bandwidth costs are incurred throughout the CDN system, our focus is the cost of transmitting content from edge servers to end-users (see Figure 1) that constitutes the lion's share of the bandwidth costs in the system.

A CDN buys bandwidth from ISPs using network contracts that fall into one of three types depending on how the bill for traffic usage is computed in each billing period. The billing period (typically, a month) is divided into a sequence of M time buckets (typically, 5-minute buckets, so that there are about $M = 8640$ buckets per month). Each ISP computes the *traffic profile* $\langle b_1, b_2, \cdots, b_{M-1}, b_M \rangle$, where b_i represents the average traffic (in Mbps) sent in time bucket i from the CDN's servers located in that ISP. Then, depending on the type of the contract, the *billable traffic* for the billing period is computed as either the average (AVG), the maximum (MAX), or the 95^{th} percentile of the values $\langle b_1, b_2, \cdots, b_{M-1}, b_M \rangle$. The CDN pays the ISP the product of an agreed-upon unit cost (in dollars per Mbps) and the billable traffic (in Mbps). The unit costs vary from ISP to ISP, with some ISPs being cheaper than others, depending on the specifics of the contracts negotiated between the CDN and the ISPs. Note that while most real-world contracts are either AVG or 95^{th}, MAX is highly important from a practical system

[1] A CDN also incurs fixed costs such as costs for servers and colocation.

design perspective, since traffic cannot be controlled in a precise enough fashion[2] to take advantage of the 5% window for free traffic in a 95^{th} percentile contract. Therefore, real-life bandwidth cost optimizers view 95^{th} percentile contracts as MAX contracts for purposes of the optimization, and hence studying the MAX contract model is very important.

The overall bandwidth cost incurred by a CDN is the sum of the costs incurred at each individual ISP, where each ISP charges the CDN for traffic usage as specified in the contract. Because the contract terms with each ISP can vary significantly, the manner in which the CDN splits up the aggregate traffic demand between individual ISPs significantly influences the overall bandwidth cost incurred by the CDN. The primary focus of this paper is optimal algorithms for assigning traffic demand across multiple ISPs to minimize the overall bandwidth cost. In particular, we seek algorithms that produce solutions that are *provably* optimal or near-optimal.

1.3. Mirroring and Multihoming

While the model and results presented in this paper use CDNs as a motivating example, the results are also applicable to other important technologies such as multihoming [2], where an enterprise contracts with multiple ISPs to provide redundant Internet access for its origin infrastructure. The enterprise would then route traffic to and from its origin via uplinks that connect to the Internet via different ISPs, so as to minimize bandwidth costs and maximize availability and performance. A multihomed enterprise can use a number of techniques to manage the traffic on its uplinks. The enterprise can manage multiple ip address spaces associated with multiple ISPs and use the Domain Name System (DNS) to resolve each domain name to an appropriate ip address. The routes used by traffic to that domain name is governed by the ip address that is returned by DNS. Alternately, enterprises may manage a single ip address space and use the Border Gateway Protocol (BGP) to appropriately announce all or portions of this address space on the various uplinks, thereby controlling the traffic routes through those links [16]. In addition to multihoming, the enterprise could also create multiple replicas (or, mirrors) of its origin infrastructure in different ISPs and different geographies. Multihoming and mirroring are used by large enterprises in a complementary fashion to using a CDN. Our model and results are also applicable to the problem of assigning the origin traffic to multiple mirrors and/or multihomed uplinks to minimize bandwidth cost.

1.4. Performance versus Cost

While this paper considers optimizing cost in isolation, real-world technologies such as CDNs and multihoming aim to first optimize a notion of performance (such as minimizing web download time by reducing latency and loss) while striving to optimize cost.

[2] The imprecision comes from several sources. For instance, some browsers don't comply with TTLs in a precise fashion, and traffic moved away from an ISP by the optimizer will decay slowly over time instead of falling sharply.

However, pure cost optimization that we consider in this paper is an important first step for the following reasons.

- From an algorithmic standpoint understanding pure cost optimization is a major stepping stone for the more general bi-criteria cost-performance optimization that we plan to do in future work. We believe that the algorithmic ideas generated in this study will shed light on the more complex bi-criteria optimization framework.
- Different types of traffic have different sensitivities to performance and cost. Delivering a real-time application is extremely performance sensitive but also less cost sensitive as customers are willing to pay more for higher performance. However, other types of traffic such as (non-realtime) background downloads of large files is less performance sensitive but also more cost sensitive as customers expect to pay much less. The latter situation is more closely aligned with the pure cost optimization regime presented in this paper.
- The pure cost optimization studied in this paper provides a lower bound on the bandwidth cost achievable by any real-world system that simultaneously optimizes performance and cost. Comparing the actual incurred cost with this lower bound delineates the portion of the actual cost that is intrinsic to the contracts and traffic from the remaining additional cost premium attributable to providing performance and other considerations. Understanding this cost premium and how it varies with different types of traffic is critical to understanding the cost structure of the content delivery service.

1.5. Prior Work

Considering the practical importance of the problem in recent years, heuristic implementations exist. However, this is the first formal study[3] of algorithms for bandwidth cost minimization across multiple ISPs. Recently, there has been some interesting work on cost minimization from a multihoming perspective [14] where AVG and 95^{th} percentile contracts are considered and empirically evaluated. However, our work is unique in considering the typical CDN situation where the optimizer *simultaneously* routes traffic to ISPs with *bounded* capacities and a mix of contract types, and formal bounds for optimality are shown in the competitive ratio framework for online algorithms. There is extensive literature on online algorithms [12, 13]. Prior research on online algorithms for ski-rental and related problems [10, 9] is particularly relevant as we show interesting connections between our problem and this class of problems. Specifically, our techniques to solve the bandwidth minimization problem are inspired by those used to solve variants of the ski-rental problem. In addition, the competitive ratios of our online algorithms are also reminiscent of those derivable for the ski rental problem.

[3] A preliminary version of this paper appeared as [15].

CDNs have been the focus of much research in recent years [17, 18], though there has not been much prior work in the current context of bandwidth cost optimization. In our work, we primarily consider a CDN that owns and operates a dedicated distributed network of servers that deliver content on behalf of content providers. The dedicated network approach is the predominant model for CDNs today with providers such as Akamai and Limelight utilizing that model. However, there are other models for content delivery that have been proposed. For instance, multiple CDNs could cooperate to serve content [25]. Alternately, content delivery can be achieved by P2P systems such as KaZaa [20], or Gnutella [19] that utilize (non-dedicated) peers to serve other peers. High-quality content delivery is harder to achieve with P2P systems in practice, though there has been has been much recent research to make P2P systems more scalable, more available and better performing, including a number of experimental systems such as Chord [21], Content Addressable Networks [22], Tapestry [23], and Pastry [24]. Bandwidth cost is less of an issue for P2P systems since peer bandwidth is typically free from the perspective of the entity that provides the P2P service. However, much of the enterprise-quality content delivery happens today on traditional CDNs in the dedicated network model where our research is directly applicable.

1.6. 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 future algorithmic investigation. The model and algorithms presented here are immediately relevant to commercial technologies of today, advancing the current state-of-the-art. Our goal here is to develop algorithmic techniques for cost optimization and to derive *provably* optimal algorithms, leaving the empirical study of these ideas for future work.

We study both *offline* and *online* algorithms for the bandwidth cost minimization problem. An offline algorithm knows the traffic that needs to be routed for the entire billing period in advance. While an online algorithm makes routing decisions knowing only the past and current traffic levels and without any knowledge of the future traffic. Both kinds of algorithms are useful in practice. Routing traffic in an actual system is necessarily online, while offline algorithms are used for retrospective cost analysis. In Section 3, 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. Note that the offline optimal algorithm produces a lower bound on the cost against which any online algorithm can be compared at the end of each billing period. Further, an optimal offline algorithm is of independent interest in practice since it can be used retrospectively to derive the lowest achievable cost of the prior billing period. A comparison of the offline optimal cost with the (higher) actual cost incurred during the billing period provides valuable information on the cost structure of the CDN, i.e., which portion of the cost is an inevitable function of the ISP contracts and what is the additional cost for providing greater performance in an online setting.

Next, in Section 4, we turn to online algorithms that know only the current and the past traffic levels, and are unaware of any events in the future. Note that any cost optimizer that is implemented as a part of the mapping system is necessarily online. Specifically, in Section 4.1, we devise a deterministic online algorithm that is at most a factor of 2 in cost from the optimal offline solution. Further, in Section 4.2, we devise a randomized online algorithm that has an expected cost that is a factor of at most $\frac{e}{e-1}$ from the optimal offline solution. In both cases, we show that the competitive ratios are the best possible. Note that our results fully characterize the value of knowing future traffic in bandwidth cost minimization. Another interesting theoretical contribution of this work is that we show intriguing connections between the online bandwidth optimization problem and the seemingly-unrelated but well-studied ski rental problem. Specifically, our work shows that the online decision to route through a MAX versus an AVG contract is a generalized form of the buy-versus-rent decision in the ski-rental problem. This furthers our understanding of the class of online problems where competitive ratios of 2 and $e/(e-1)$ are optimal for deterministic and randomized online algorithms respectively. Other problems in this class include previously known generalizations of ski rental, such as the Bahncard problem [7] and the TCP Acknowledgment problem [6, 9] where the same competitive ratios apply.

In Section 5, we extend the contract framework to include the notion of a committed information rate (CIR), where the CDN has paid in advance for a certain committed amount of traffic through an ISP. We extend our results of Section 3 to provide an optimal offline algorithm for MAX and AVG contracts with CIR.

Finally, we show the intractability of optimizing 95^{th} percentile contracts. Specifically, we show that optimizing costs for 95^{th} percentile contracts is NP-hard, differentiating it from the MAX and AVG contracts.

2. The Bandwidth Cost Minimization Problem

In this section, we model network contracts and formally describe the bandwidth cost minimization problem.

2.1. Network Contracts

A first important step in our study is accurately modeling the parameters of a CDN's typical 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.

1. Type. The contract type dictates how the ISP will bill for the traffic that is sent over its links. As noted earlier in Section 1.2, the three types of contracts are AVG, MAX, and 95th.
2. Unit Cost. *Unit cost* C is the cost per Mbps that the ISP charges the CDN.

3. Capacity. The capacity P is the maximum bandwidth (in Mbps) that one can transmit from the CDN's servers in that ISP.

The amount that a CDN pays ISP_j is computed using the first two parameters above as shown below. The billing period (typically, a month) is divided into a sequence of M time buckets (typically, 5-minute buckets, so that there are about $M = 8640$ buckets per month).

1. The traffic profile $\left\langle y_1^j \cdots, y_t^j, \cdots y_M^j \right\rangle$ is computed, where y_t^j represents the average traffic sent (in Mbps) in time bucket t from the CDN's servers located in ISP_j. Next, the *sorted traffic profile* $\left\langle x_1^j, x_2^j, \cdots, x_{M-1}^j, x_M^j \right\rangle$ is computed by sorting the traffic profile in descending order, i.e., $x_1^j \geq x_2^j \geq \cdots \geq x_M^j$.
2. For an AVG contract the billable traffic is computed to be $t_j = \sum_i x_i^j / M$. Likewise, the billable traffic of a MAX contract is $t_j = x_1^j$. And, the billable traffic of a 95^{th} contract is $t_j = x_{\frac{M}{20}}$, since $\frac{M}{20}$ time periods represents 5% of the billing period and hence $x_{\frac{M}{20}}$ is the 95^{th} percentile of the traffic values in the billing period.
3. The total amount that the CDN pays the ISP_j is the unit cost C_j (in dollars per Mbps) multiplied by the billable traffic t_j (in Mbps).

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. CIRs are considered in the later part of the paper in Section 5.

2.2. Cost Minimization

The optimization problem proposed here models an aspect of the mapping component in a CDN that senses the incoming traffic requests and assigns them to servers in multiple ISPs. Typically, the traffic assignment is performed by resolving domain names using DNS, and the incoming traffic represents requests from thousands of nameservers around the world. For simplicity, we will assume that the total traffic demand as well as the traffic routed through each ISP during each time interval are integers. Further, we assume that the traffic can be split and assigned in any manner to the ISPs at the granularity of a single unit of traffic. This is a good first-cut approximation as most of the Internet web traffic comes from a large number of nameservers. Each nameserver can be routed independently by responding to the DNS request from the nameserver with an appropriate set of server ips. Since each nameserver is responsible for only a small portion of the total traffic, it is possible to control the routing of the traffic at a fine granularity.

The bandwidth cost minimization problem is modeled as follows. The billing period (typically one month) is divided into M 5-minute time buckets. We model the incoming aggregate traffic demand as a sequence $I = \langle b_1, \cdots, b_t, \cdots b_M \rangle$, where b_t is the average

traffic (Mbps) in time bucket t. Note that b_t represents the total traffic demand from end-users that must be served by the CDN at time bucket t. 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. The total cost incurred by the traffic routing algorithm for the input traffic sequence I is simply

$$C(I) = \sum_j C_j t_j,$$

where C_j is the unit cost of ISP_j and the t_j is the billable traffic computed from the profile $\left\langle y_1^j \cdots, y_t^j, \cdots y_M^j \right\rangle$ of traffic served from ISP_j taking into consideration the type of contract.

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. It makes traffic routing decisions based on this complete knowledge. An online algorithm makes routing decisions at time t knowing only b_j, $1 \leq j \leq t$, i.e., knowing only the past and current values. Note that the incoming traffic b_t, the traffic assignments y_t^j, and capacities P_j are *integral* values in the units of bits per second.

As mentioned earlier, 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 [13] 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. Note that the competitive ratio guarantees derived for our online algorithms hold in the worst-case, irrespective of the behavior and (un)predictability of the incoming traffic.

3. The Offline Algorithm

In this section, we derive an optimal offline algorithm that routes traffic with minimum total bandwidth cost to ISPs with AVG or MAX contracts. Without loss of generality, we assume that no two AVG ISPs (resp., MAX ISPs) have the same unit cost, since two such ISPs can be considered to be one ISP with the sum of their individual capacities.

3.1. MAX ISPs

To start with, assume that we are given contracts that are all MAX ISPs and there are no AVG ISPs. Let there be m MAX ISPs Max_i, $1 \leq i \leq m$, such that $C_{Max_1} < C_{Max_2} < \cdots < C_{Max_m}$. Define the threshold t_{Max_i} of an ISP Max_i to be the maximum traffic routed during the billing period through that ISP. The following lemmas hold.

Lemma 1. *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 that 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. □

Lemma 2. *There exists an optimal solution in which Max_i is not used in a time interval unless 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 1 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. □

Thus the greedy algorithm of using a cheaper MAX ISPs to its full capacity before using a costlier MAX ISPs routes traffic through m MAX ISPs with the least cost. As the cost is determined by the bucket with most traffic to be routed the time taken to calculate the cost of the optimal routing is $O(m \log m + M)$, since sorting the contracts by cost takes $O(m \log m)$ time and finding the bucket with maximum traffic takes $O(M)$ time.

3.2. AVG ISPs

Now we give a similar greedy algorithm for routing traffic when we have only AVG ISPs. Assume that we are given contracts from n AVG ISPs Avg_i, $1 \leq i \leq n$, such that $C_{Avg_1} < C_{Avg_2} < \cdots < C_{Avg_n}$. The following Lemma holds.

Lemma 3. *In any optimal solution, ISP Avg_i is not used in a time interval 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. □

Thus the greedy algorithm where in each interval a cheaper AVG ISPs is used to its full capacity before using costlier AVG ISPs routes traffic through n AVG ISPs with the least cost. We can find the most expensive AVG ISP that needs to be used in a bucket in $O(\log n)$ time by using binary search to search for the bucket capacity in an array of size n, whose k^{th} element is $\sum_{i=1}^{k} C_{Avg_i}$ for $1 \leq k \leq n$. As the ISPs need to be sorted by their cost, the total time taken to calculate the cost of the optimal solution is $O((n + M) \log n)$.

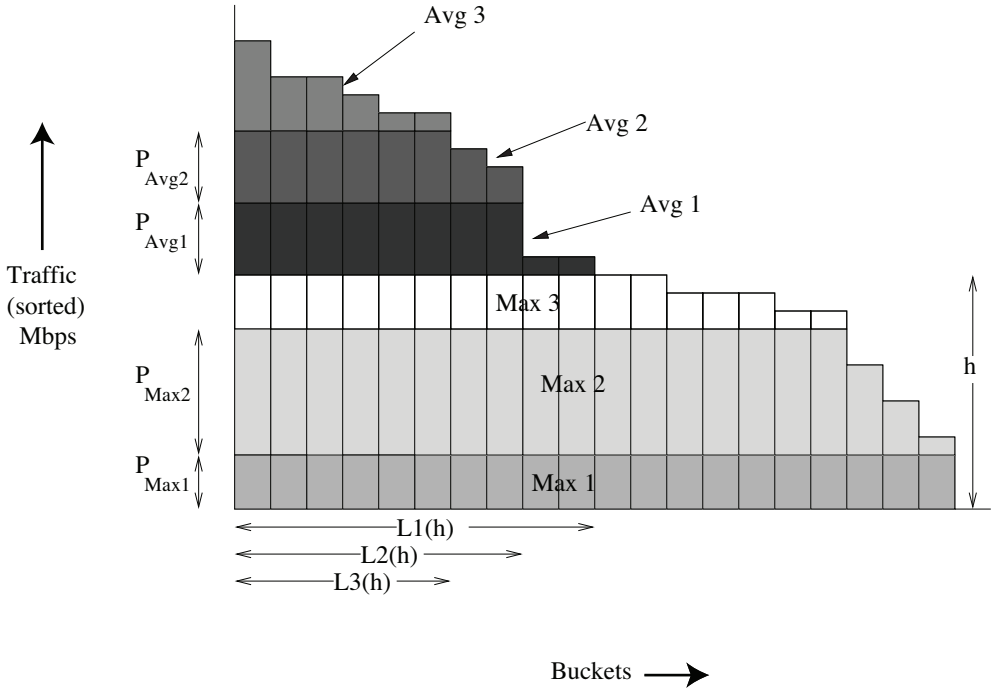

Figure 2: The structure of an optimal offline solution

3.3. MAX ISPs and AVG ISPs

Now we consider the general case when we have *both* MAX ISPs and AVG ISPs. 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 also given contracts from n AVG ISPs Avg_i, $1 \leq i \leq n$, such that $C_{Avg_1} < C_{Avg_2} < \cdots < C_{Avg_n}$.

Let $x_1 \geq x_2 \geq \cdots \geq x_M$ be the average traffic within each of the M time buckets during the billing period, sorted and placed in descending order. The assignment of traffic to ISPs over a billing period can be represented visually as in Figure 2. The vertical bars represent the x_i, $1 \leq i \leq M$. Each vertical bar is subdivided horizontally to represent the assignment of that traffic to multiple ISPs. We now show that there exists an optimal solution that is of the form shown in Figure 2.

Lemma 4. *There exists an optimal solution such that in any time interval an AVG ISP is used only if all MAX ISPs are used to their respective thresholds for the billing period.*

Proof: Start with any optimal solution where ISP Avg_i receives $x > 0$ units of traffic in a time interval, but some ISP Max_j is used less than its threshold 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 does not increase while the cost of Max_j remains the same. Thus, the overall cost does not increase and we have an optimal solution which satisfies the given property. □

Thus there exists a dividing line at some height h (Figure 2) such that all traffic below this line is routed through MAX ISPs and all traffic above is routed through AVG ISPs.[4] Thus the problem can be broken into three parts - finding the optimal height h of the dividing line, routing traffic below the height h through MAX ISPs and routing traffic above the height h through AVG ISPs. The problem of routing traffic below (resp., above) the dividing line at height h through only MAX ISPs (resp., AVG ISPs) can be solved by greedy algorithms given in Section 3.1 (resp. Section 3.2). The Max-Threshold h, defined to be the sum of the thresholds of the MAX ISPs, can be found by binary search using the following lemma.

Define $C_{Max}(h)$ (resp., $C_{Avg}(h)$) to be the total cost of routing traffic below (resp., above) the dividing line at height h through the MAX ISPs (resp., AVG ISPs) using the greedy algorithms given above. Since $C_{Max}(h)$ (resp., $C_{Avg}(h)$) is right differentiable, we define $C'_{Max}(h^+)$ (resp., $C'_{Avg}(h^+)$) to be its right derivative at h i.e., $\lim_{\delta h \to 0^+}(C_{Max}(h+\delta h) - C_{Max}(h))/\delta h$. Let $C(h) = C_{Max}(h) + C_{Avg}(h)$ and thus $C'(h^+) = C'_{Max}(h^+) + C'_{Avg}(h^+)$. Further, recall that x_1 the largest traffic that is to be routed within any time bucket.

Lemma 5. *For all h_1,h_2 if $C'(h_1^+)$ and $C'(h_2^+)$ are well-defined and $x_1 - \sum_{i=1}^{n} P_{Avg_i} \leq h_1 < h_2 < \sum_{j=1}^{m} P_{Max_j}$, then $C'(h_1^+) \leq C'(h_2^+)$.*

Proof: $C'_{Max}(h^+)$ is the cost of the cheapest MAX ISP that has not been used to its full capacity when the Max-Threshold is h. Thus $C'_{Max}(h^+)$ is defined wherever $C_{Max}(h)$ is defined, except when h is the sum of the capacities of the MAX ISPs. From Lemma 2, it follows that $C'_{Max}(h^+)$ is a non-decreasing function.

$C'_{Avg}(h^+) = -\sum_{i=1}^{M}$(cost of the most expensive AVG ISP used in the i^{th} interval when the Max-Threshold is h). Thus $C_{Avg}(h)$ is right differentiable wherever it is defined. From Lemma 3, it follows that $C'_{Avg}(h^+)$ is a non-decreasing function. The lemma follows as $C'(h^+) = C'_{Max}(h^+) + C'_{Avg}(h^+)$.□

Given m MAX ISPs, n AVG ISPs, and the traffic values for the entire billing period, the offline optimal algorithm (which we refer to as OPT) works as follows:

1. Using binary search, compute the optimal Max-Threshold h as the value that minimizes the cost function $C(h)$.
2. Route all traffic at or below h greedily through the MAX contracts.
3. Route all traffic above h greedily through AVG contracts as shown in Figure 2.

Theorem 6. *The offline optimal solution and its cost can be computed in $O(L(\log m + M \log n) + n \log n + m \log m)$ time, where m is the number of MAX ISPs, n is the*

[4]Note that the algorithm could produce a solution that uses only the AVG contracts, if that is optimal, by computing the height h to be zero. In fact, that would be the case if the unit cost of the AVG contracts are significantly lower than the unit cost of the MAX contracts.

number of AVG ISPs, M is the total number of intervals in the billing period and L is the number of bits required to represent the maximum amount of traffic sent in an interval.

Proof: $C(h)$ is a continuous function as both $C_{Max}(h)$ and $C_{Avg}(h)$ are continuous functions. From Lemma 5 and the fact that $C(h)$ is continuous it follows that $C(h)$ reaches its minimum whenever $C(h^+)$ changes from being non-positive to being positive. As a preprocessing step, we sort the MAX ISPs and AVG ISPs in the ascending order of their unit cost in time $O(n \log n + m \log m)$. Then, we use binary search over all values of h to find the h such that $C(h^+)$ changes sign. There are at most 2^L possible values for h. So there are at most $\log(2^L) = L$ steps. At each step of the binary search we need to calculate $C'(h^+)$ by computing $C(h)$ values. This can be done in $O(\log m + M \log n)$ time, using binary search to find the most expensive AVG ISP used in each bucket and the most expensive MAX ISP used. The most expensive MAX ISP to be used can be found in $O(\log m)$ time by using binary search to search for h in an array of size m, whose k^{th} element is $\sum_{i=1}^{k} C_{Max_i}$ for $1 \leq k \leq m$. Thus we can calculate the optimal solution and its cost in $O(L(\log m + M \log n) + n \log n + m \log m)$ time.□

4. Online Algorithms

We provide both deterministic and randomized optimal online algorithms for the problem of routing traffic through AVG and MAX ISPs with minimum cost with competitive ratios of 2 and $\frac{e}{e-1}$ respectively. Note that an online algorithm at time t knows the current and past traffic values, $b_1, b_2, \cdots, b_t$, but does not know future traffic values $b_{t+1}, b_{t+2}, \cdots, b_M$.

4.1. Optimal Deterministic Online Algorithm

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

1. Runs the offline algorithm OPT of Section 3 on the input $\langle b_1, b_2, \cdots, b_t, 0, 0, \cdots, 0 \rangle$. That is, run the optimal offline algorithm on a prefix of the input assuming all future time intervals have zero traffic.
2. Routes the current traffic b_t in the same manner as OPT.

Note that running OPT in step 1 at time t results in an optimal Max-Threshold h_t being computed. First, we show that the Max-Thresholds h_t can only increase with time t as we progress through the billing period.

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

Proof: Assume $h_t > h_{t+1}$. The cost of routing the traffic $\langle b_1, b_2, \cdots, b_t, 0, 0, \cdots 0\rangle$ with a Max-Threshold of h_t is less than or equal to cost of routing the same traffic with a Max-Threshold of h_{t+1}. As $b_{t+1} - h_t < b_{t+1} - h_{t+1}$ the contribution in the total cost of routing the $t + 1^{th}$ interval traffic above the Max-Threshold through the AVG ISPs with a Max-Threshold of h_t is less than or equal to the same with a Max-Threshold of h_{t+1}. Thus with a Max-Threshold of h_t we can route the traffic $\langle b_1, b_2, \cdots, b_t, b_{t+1}, 0, 0, \cdots 0\rangle$ with the same or lower cost than with a Max-Threshold of h_{t+1}. This contradicts the fact that for no Max-Threshold of $h > h_{t+1}$ can we route the traffic $\langle b_1, b_2, \cdots, b_t, b_{t+1}, 0, 0, \cdots 0\rangle$ with the same or lower cost. Hence proved by contradiction.□

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. Also, by Lemma 7, $h_M \geq h_t$, for all $t \leq M$. Therefore,

$$C_{A,Max} = C_{OPT,Max} \leq C_{OPT} \tag{1}$$

Let $C^t_{A,Avg}$ be the cost incurred in AVG ISPs by algorithm A during the first t time intervals. 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 $C^t_{A,Avg} \leq C^t_{OPT}$.
Base Case: When $t = 1$, algorithm A runs OPT on the first input and behaves identical to it. Therefore,

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

Inductive Case: Assume that the hypothesis is true until t, i.e., $C^t_{A,Avg} \leq C^t_{OPT}$. As C^t_{OPT} is the cost of the optimal offline solution for input $\langle b_1, b_2, \cdots, b_t, 0, 0, \cdots, 0\rangle$, we have that $C^t_{OPT} \leq C'$, where C' is the cost of the solution for the same input $\langle b_1, b_2, \cdots, b_t, 0, 0, \cdots, 0\rangle$ but using a Max-Threshold of h_{t+1} . Therefore, it follows that

$$C^t_{A,Avg} \leq C'. \tag{2}$$

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 both sides of Equation 2, we get $C^{t+1}_{A,Avg} \leq C^{t+1}_{OPT}$. This completes the induction. Therefore,

$$C_{A,Avg} = C^M_{A,Avg} \leq C^M_{OPT} = C_{OPT} \tag{3}$$

Thus, combining Equations 1 and 3,

$$C_A = C_{A,Max} + C_{A,Avg} \leq 2C_{OPT}.$$

□

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

Proof: We first prove that the Ski Rental problem [9] is a special case of the bandwidth cost minimization problem. Given a ski rental problem where the cost of renting a pair of skis is 1 and the cost of buying them 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 with unit cost p and one AVG ISP with unit cost M, where $M \geq k$ is the number of intervals in the billing period. The input traffic $b_t = 1$ unit, if $1 \leq t \leq k$, and zero for $k < t \leq M$. The capacity of each ISP is 1 unit. Thus in an interval one can only send either 0 or 1 unit of traffic through an ISP, since traffic values are integral. By our transformation, buying skis is the optimal strategy for original ski rental problem if and only if the optimal solution for traffic routing problem is to use only the MAX ISP for routing the entire traffic. Similarly, renting skis is optimal if and only if AVG ISP is used to route the entire traffic in the optimal solution. Also the value of the optimal cost in both problems is the same.

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 [9, 10] on the competitive ratio of a deterministic online algorithm of $1 + \frac{\lceil p \rceil - 1}{p}$ which $\longrightarrow 2$ as $p \longrightarrow \infty$. □

4.2. Optimal 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.

If the time-ordered sequence of traffic demands is $I = \langle b_1, b_2, \cdots, b_M \rangle$ then at a given time interval t, the deterministic online algorithm A_z does the following:

1. Runs the offline algorithm $OPT(z)$ of Section 3 on 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. Routes the current traffic b_t in same manner as $OPT(z)$.

Note that A_1 is the deterministic online algorithm A given in Section 4.1. 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 for the costs of the MAX ISPs. The proofs of the following two lemmas are similar to the analogous proofs of Equations 1 and 3 in Theorem 8.

Lemma 10. $zC_{A_z,Max} = C_{OPT,Max}(z)$

Proof: This is proved in the same fashion as Equation 1 in Theorem 8 where we showed that $C_{A,Max} = C_{OPT,Max}$. 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. The lemma follows.□

Lemma 11. $C_{A_z,Avg} \leq C_{OPT}(z)$

Proof: This proof is similar to the inductive proof given for Equation 3 in Theorem 8 where we showed that $C_{A,Avg} \leq C_{OPT}$. □

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 $0 \leq z \leq v \leq 1$,

$$\begin{aligned} C_{OPT}(v) &= C_{OPT,Max}(v) + C_{OPT,Avg}(v) \\ &= vC_{A_v,Max} + C_{OPT,Avg}(v) \\ &\quad \text{(using Lemma 10)} \\ d(C_{OPT}(v)) &= dv \cdot C_{A_v,Max} + v \cdot d(C_{A_v,Max}) \\ &\quad + d(C_{OPT,Avg}(v)) \end{aligned} \tag{4}$$

Define $h(w)$ to be the Max-Threshold in the optimal offline solution with cost $C_{OPT}(w)$ 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 most expensive MAX ISP that was used in optimal offline solution with cost $C_{OPT}(w)$ (or in algorithm A_w with cost C_{A_w}). As the actual cost of any MAX ISP used in the gap between $h(v+dv)$ and $h(v)$ would be at most C_{Max_v}, the increase in cost of MAX ISPs when Max-Threshold is increased from $h(v+dv)$ to $h(v)$ is at most $C_{Max_v} \cdot (h(v) - h(v+dv))$. Thus,

$$\begin{aligned} -d(C_{A_v,Max}) &= C_{A_v,Max} - C_{A_{v+dv},Max} \\ &\leq C_{Max_v} \cdot (h(v) - h(v+dv)) \\ &= -C_{Max_v} \cdot d(h(v)) \end{aligned} \tag{5}$$

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 $h(v)$ to $h(v+dv)$ decreases the cost

due to the MAX ISPs by at least $vC_{Max_{v+dv}} * (h(v) - h(v+dv))$. The corresponding increase in the cost due to the AVG ISPs is at least the decrease in cost due to the MAX ISPs, since $C_{OPT}(v)$ is the optimal cost. Thus,

$$\begin{aligned} d(C_{OPT,Avg}(v)) &= C_{OPT,Avg}(v+dv) \\ &\quad -C_{OPT,Avg}(v) \\ &\geq vC_{Max_{v+dv}} \\ &\quad \cdot(h(v) - h(v+dv)) \\ &= -vC_{Max_{v+dv}} d(h(v)) \end{aligned} \tag{6}$$

Substituting Equations 5, and 6 in Equation 4, we obtain

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

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 $C_{OPT}(1) - C_{OPT}(z) \geq \int_z^1 C_{A_v,Max} dv$. □

Corollary 13. $C_{OPT}(1) \geq \int_0^1 C_{A_w,Max} dw$

Proof: Follows from Lemma 12 by setting $z = 0$.□

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

$$\begin{aligned} C_{A_z} &= C_{A_z,Max} + C_{A_z,Avg} \\ &\leq C_{A_z,Max} + C_{OPT}(z) \\ &\quad \text{(by Lemma 11)} \\ &\leq C_{A_z,Max} + C_{OPT}(1) \\ &\quad - \int_z^1 C_{A_w,Max} dw \\ &\quad \text{(by Lemma 12)} \end{aligned}$$

$$
\begin{aligned}
E[C_{A_{Rand}}] &= \int_0^1 C_{A_z} p(z) dz \\
&\leq C_{OPT}(1) + \int_0^1 C_{A_z,Max} p(z) dz \\
&\quad - \int_0^1 p(z) (\int_z^1 C_{A_w,Max} dw) dz \\
&= C_{OPT} + \int_0^1 C_{A_z,Max} p(z) dz \\
&\quad - \int_0^1 C_{A_w,Max} (\int_0^w p(z) dz) dw \\
&= C_{OPT} + \\
&\quad \int_0^1 (p(z) - P(z)) C_{A_z,Max} dz
\end{aligned}
$$

$$
\text{Competitve Ratio} = \frac{E[C_{A_{Rand}}]}{C_{OPT}} \leq 1 + \frac{\int_0^1 (p(z) - P(z)) C_{A_z,Max} dz}{\int_0^1 C_{A_z,Max} dz}
$$

(by Corollary 13)

Setting $p(z) = \frac{e^z}{e-1}$ and $P(z) = \frac{e^z - 1}{e-1}$, the RHS of the above equation is equals $1 + 1/(e-1) = e/(e-1)$. □

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 the bound for the ski rental problem similar to the randomized online algorithm for the snoopy caching problem [10].) 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. Extensions

In this section, we consider two different extensions to our results. In Section 5.1 we consider the notion of Committed Information Rate (CIR) and in Section 5.2 we consider 95^{th} percentile contracts.

5.1. Committed Information Rate (CIR)

Committed Information Rate (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. Traffic sent over and above the CIR is called the burst rate and is charged for in proportion to usage at a specified unit cost. Since the cost for the CIR is prepaid regardless of usage, it can be assumed that traffic up to the CIR can be routed with an incremental cost of 0. Let $x_1 \geq x_2 \geq \cdots \geq x_M$ be the average traffic within each of the M 5-minute intervals during the billing period, placed in descending order. For an AVG contract, the bill for the month is $C_{AVG} * (\sum_i x_i/M - CIR_{AVG})$ if $\sum_i x_i/M \geq CIR_{AVG}$, otherwise it is 0 , where C_{AVG} is the unit cost and CIR_{AVG} is the CIR of the AVG ISP. For a MAX contract, the bill for the month is $C_{MAX} * (x_1 - CIR_{MAX})$ if $x_1 \geq CIR_{MAX}$, otherwise it is 0, where C_{MAX} is the unit cost and CIR_{MAX} is the CIR of the MAX ISP. Similar we can postulate the cost function for a 95^{th} percentile contract that incorporates CIR.

5.1.1. Offline algorithm

We derive an offline algorithm for routing through AVG and MAX ISPs with CIR. First, in Section 5.1.2, we consider routing through MAX ISPs in isolation. Then, in Section 5.1.3, we consider routing in AVG ISPs in isolation. Finally, in Section 5.1.4, we combine the two approaches to produce an offline optimal algorithm when we have both MAX and AVG ISPs.

5.1.2. MAX ISPs with CIR

Assume that we are given contracts from m MAX ISPs Max_i, $1 \leq i \leq m$, with unit cost C_{Max_i}, capacity P_{Max_i} and CIR CIR_{Max_i} ($\leq P_{Max_i}$), such that for all $j < i$, $C_{Max_j} \leq C_{Max_i}$. Further, assume that there are no AVG ISPs.

Lemma 16. *In any optimal solution, threshold $t_{Max_i} > CIR_{Max_i}$ 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: The proof is similar to the proof of Lemma 1. □

Lemma 17. *There exists an optimal solution in which Max_i is not used more than its CIR in a time interval unless each ISP Max_j, $j < i$, has been used to its full capacity of P_{Max_j} and all MAX ISPs have been used at least to their CIR.*

Proof: The proof is similar to the proof of Lemma 2. □

Lemma 17 above gives us an optimal greedy algorithm for routing traffic through MAX ISPs alone. First use the CIRs of all the MAX ISPs and then route the remaining greedily by using cheaper ISPs to their full capacity before using costlier ISPs. This greedy algorithm also takes at most $O(m \log m + M)$ time to calculate the minimum cost of routing through m MAX ISPs.

5.1.3. AVG ISPs with CIR

Suppose we had only AVG ISPs and no MAX ISPs. We propose the following greedy algorithm for routing traffic through n AVG ISPs. Assume that we are given contracts from n AVG ISPs Avg_i with CIR CIR_{Avg_i} and capacity P_{Avg_i}, $1 \leq i \leq n$, such that $C_{Avg_1} < C_{Avg_2} < \cdots < C_{Avg_n}$. Start with the costliest AVG ISP (i.e., Avg_n) and iterate through the following steps in the decreasing order of cost, i.e., in the order $Avg_n, Avg_{n-1}, \ldots, Avg_1$, till all the traffic is routed.

Step 1. Let x_i, $1 \leq i \leq M$, be the traffic that remains to be routed in time bucket i. Let Avg_k be the current ISP under consideration, i.e., traffic has already been routed through ISPs $Avg_j, k < j \leq n$. In each time bucket i, all the traffic $(x_i - \sum_{j=1}^{k-1} P_{Avg_j})$ that cannot be routed through cheaper ISPs due to capacity constraints is routed through ISP Avg_k.

Step 2. If ISP Avg_k has not been utilized to its CIR after Step 1 above, then select the time interval i with the maximum remaining traffic (i.e., maximum traffic that is yet to be routed). If there is capacity left at time interval i in ISP Avg_k, then route an additional unit of traffic through ISP Avg_k at that time interval. Repeatedly perform this operation until either the CIR of Avg_k is exhausted or there is no more traffic to route.

Theorem 18. *The greedy algorithm given above routes traffic through n AVG ISPs with minimum cost.*

Proof: Given any other solution S, we prove that the cost of S is at least the cost of G, where G is the solution produced by our greedy algorithm. This would imply that our greedy solution is optimal. To this end, we transform solution S to solution G using a series of steps where traffic is rerouted in each step *without* increasing the total cost. Start with the costliest ISP Avg_k where traffic is routed differently in the two solutions S and G. All the costlier ISPs Avg_i, such that $i > k$, are not considered as traffic is routed through these ISP's in an identical fashion in S and G. There are 3 cases to consider.

1. If the total amount of traffic routed through ISP Avg_k in the billing period is more in solution G than in S then one can conclude that the CIR has not yet been reached forAvg_k in solution S. Hence, we can route more traffic through Avg_k in solution S till the solutions S and G route the same amount of total traffic through Avg_k. Since the additional traffic is covered by the CIR, there is no additional cost incurred in Avg_k. Thus, this traffic rerouting cannot increase the cost of solution S.
2. If the total amount of traffic routed through ISP Avg_k is less in solution G than in S, one can conclude that the additional traffic in S is above and beyond the CIR for Avg_k. Solution S can be modified to route this additional traffic that is being paid for at the higher cost of Avg_k through other ISPs Avg_i, $i < k$, that have

lower costs. This traffic rerouting to modify S cannot increase the cost of solution S.

3. If the total amount of traffic routed through ISP Avg_k is the same in solutions G and S but the manner in which the traffic is distributed over the time intervals is different, we can redistribute the traffic such that the traffic routed in all time intervals through the ISP Avg_k is the same in both solutions without increasing the total cost as follows. Note that there exists a time interval t_1 (resp., t_2) in which more (resp., less) traffic is routed through ISP Avg_k in solution G than in S. This implies that less (resp. more) traffic is routed in time interval t_1 (resp., t_2) in aggregate across ISPs $Avg_1, Avg_2, \ldots, Avg_{k-1}$ in solution G than in S. Let k_i (resp., k'_i) denote the traffic routed through ISP k at time t_i in solution G (resp., S), where $i = 1, 2$. Likewise, let h_i (resp., h'_i) denote the total traffic routed in aggregate across ISP's $Avg_1, Avg_2, \ldots, Avg_{k-1}$ at time t_i in solution G (resp. S), where $i = 1, 2$. From our discussion above, $k_1 > k'_1$ and $k_2 < k'_2$, which implies that $h_1 < h'_1$ and $h_2 > h'_2$. Note that the greedy algorithm iteratively allots traffic to ISP k in the time interval with the most remaining (i.e., unrouted) traffic, unless the capacity constraints of ISP k are reached in that time interval. Note that there is capacity remaining in ISP k at time t_2 in G, since $k_2 < k'_2 \leq P_k$. Therefore, h_2 cannot be larger than $h_1 + 1$, as otherwise the greedy algorithm would have allotted additional traffic to ISP k at time t_2 to reduce the larger remaining traffic in that time slot . Since $h_2 > h'_2$ and all values are integers, it follows that

$$h'_1 > h_1 \geq h_2 - 1 \geq h'_2 + 1 - 1 = h'_2.$$

From the inequality above we know that $h'_1 > h'_2$. Thus, there exists some ISP Avg_l, $l < k$, such that more traffic is routed through ISP Avg_l in time t_1 than in time t_2. Thus, for some $x > 0$, one can route x more units of traffic through ISP Avg_k and x less units of traffic through Avg_l in time interval t_1. And, in time interval t_2 one can route x less units of traffic through ISP Avg_k and x more units of traffic through Avg_l. This transformation does not change the total cost. But, it increases (resp., reduces) the traffic in ISP Avg_k at time t_1 (resp., t_2) in solution S, bringing the traffic pattern that is routed through ISP Avg_k in S closer to the corresponding traffic pattern in G. By repeating this process, we can make the traffic routed through Avg_k in solution S equal to the traffic routed through Avg_k in solution G for all time intervals.

Following the three steps outlined above inductively, starting from the costliest ISP and ending with the cheapest ISP, we can transform any solution S to the greedy solution G without increasing the cost. Hence, the greedy solution G is optimal. □

The time taken for the above greedy algorithm to compute the solution can be determined as follows. First, the ISP's are sorted in descending order of their unit cost, and the time intervals are sorted in the descending order of the traffic. The total time

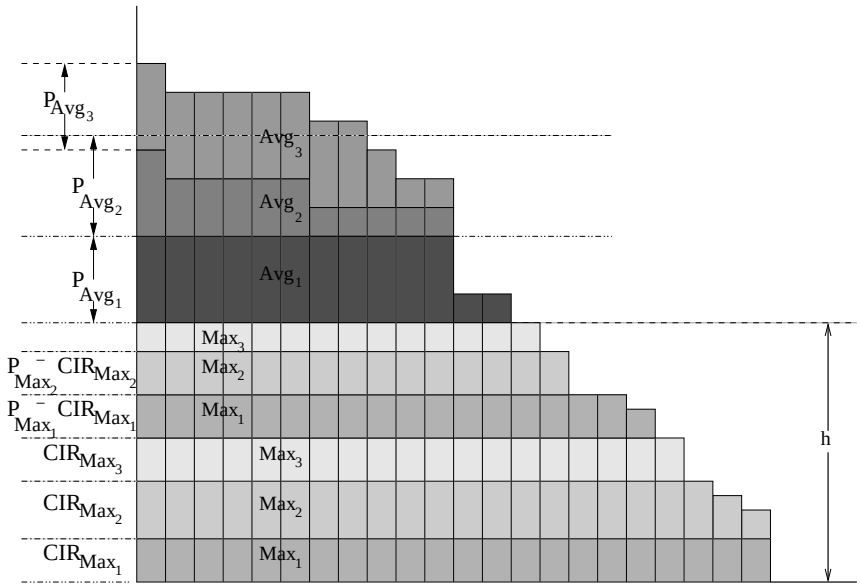

Figure 3: The structure of an optimal offline solution when the ISPs have a CIR

for the two sorts is $O(M \log M + n \log n)$. Once the time intervals and the AVG ISPs are sorted, the traffic that has to be sent through the most expensive remaining ISP Avg_k due to capacity constraints can be calculated in step 1 of the algorithm in $O(M)$ time. Routing more traffic to fully utilize the CIR of the most expensive remaining ISP Avg_k in step 2 can also be computed in $O(M + T_k)$ time, where T_k is the total amount of traffic that is routed through Avg_k. Thus routing traffic through n AVG ISPs with CIR can be done in $O(nM + T + M \log M + n \log n)$ time, where T is the total amount of traffic routed in the billing period.

5.1.4. MAX ISPs and AVG ISPs with CIR

The offline algorithm for routing traffic optimally through m MAX ISPs and n AVG ISPs, with both types of ISPs having CIR, can be visualized using Figure 3. First, a Max-Threshold h is derived. Next, traffic below threshold h is routed through the MAX ISPs using the greedy algorithm of Section 5.1.2 and the traffic above that threshold is routed through the AVG ISPs using the greedy algorithm of Section 5.1.3. The main difference between the offline algorithm for ISPs with CIR and the offline algorithm in Section 3 for ISPs without CIR is the greedy algorithm for routing traffic through AVG ISPs that is described in Section 5.1.3, since the proofs of Lemmas 4 and 5 still hold when ISPs have CIR. Similar to the proof of Theorem 6 we can prove that the cost of the optimal offline solution can be calculated in $O(L(\log m + nM) + T + n \log n + m \log m + M \log M)$ time.

Theorem 19. *The offline optimal solution (and its cost) for routing traffic in ISPs with CIR can be computed in $O(L(\log m + nM) + T + n \log n + m \log m + M \log M)$ time, where m is the number of MAX ISPs, n is the number of AVG ISPs, M is the total number of intervals in the billing period, T is the total traffic routed during the billing period, and L is the number of bits required to represent the maximum amount of traffic sent in a time interval.*

As in Section 4, it would be of interest to convert the offline optimal algorithm in this section to an online algorithm for routing in AVG and MAX ISPs with CIR. But, devising online algorithms in this context is complicated by the presence of the CIR and is a subject for future research.

5.2. 95^{th} Percentile Contracts

Unlike AVG and MAX ISPs, we now show that including network contracts that charge based on the 95^{th} percentile of the traffic renders finding the optimal offline solution NP-hard. In fact, the problem of determining whether a given input can be routed using only the free traffic (i.e., using only 5% of the intervals for each ISP) of a set of 95^{th} percentile ISPs is already NP-Hard.

Theorem 20. *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 [5], which is known to be NP-complete in the strong sense. Consider an arbitrary Bin Covering problem: We are given N positive integers as the item sizes $a_1, a_2, \ldots, a_N$, a bin capacity C, and B number of bins. We are asked whether these N numbers can be partitioned into B subsets each of which has sum at least C. The above problem instance for Bin Covering can be easily reduced to the following instance of the traffic routing problem. Given N 95^{th} percentile ISPs with capacities equal to $a_1, a_2, \ldots, a_N$ can we route the following traffic pattern with zero cost. For the first $M/20 - 1$ intervals the traffic to be routed is $\sum_{i=1}^{n} a_i$ and for the next B intervals C amount of additional traffic is to be routed. (M is chosen such that $M \geq M/20 - 1 + B$.) Note that $M/20$ intervals constitute 5% of the time intervals in the billing period. Each ISP is filled to the capacity for the first $M/20 - 1$ steps, and so each have only one additional time interval to route for free. Allocating the additional traffic into that free time interval for each ISP amounts to a solution for the original Bin Covering problem instance. □

Due to the fact that finding whether traffic can be routed with zero cost is NP-hard, unless $P = NP$, there cannot exist an approximation algorithm for the bandwidth cost minimization problem with 95^{th} percentile ISPs that produces a solution that is within a constant additive term of optimal.

6. Conclusions

An important contribution of this paper is that it opens up the algorithmically rich and practically important area of bandwidth cost optimization for CDNs and multihomed enterprises using realistic contract models. More specifically, we provided an optimal offline algorithm that routes traffic to minimize the total bandwidth cost incurred in ISPs with AVG and MAX contracts. Also, we provided deterministic and randomized online algorithms that have optimal competitive ratios. Finally, we established interesting theoretical connections between bandwidth cost minimization and the well-studied buy-versus-rent paradigm of the ski rental problem.

This paper is but a first step into this area of research, and many open questions for future research remain. Our current algorithmic work does not incorporate CIRs in an online setting. Devising near-optimal online algorithms under the right adversarial model for AVG and MAX contracts with CIR is a problem of great importance for future work. Further, devising a suitable definition of approximation and finding good approximation algorithms for 95^{th} percentile contracts is another interesting avenue for future investigation.

Finally, a critical avenue for future research is to introduce the notion of performance and extend our model and algorithms to simultaneously optimize both cost and performance. We believe that the current work provides a first step towards reaching this final objective. In addition, it is important to study the behavior of our algorithms empirically by simulating them on realistic traffic traces and network contracts. Specifically, it would be of interest to collect traffic traces from a large distributed CDN and empirically study the bandwidth cost reduction that is possible by using our algorithmic ideas. Any such study must also incorporate the performance optimization criteria to ensure that cost minimization does not adversely impact the performance of end-users.

Acknowledgements

The work of Micah Adler is supported in part by the National Science Foundation under the NSF Early Career Development Award CCF-0133664 and an NSF Award CNS-0325726. The work of Ramesh Sitaraman is supported in part by a National Science Foundation award under grant number CNS-0519894.

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] Limelight. http://www.limelightnetworks.com.

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

[5] 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.
[6] 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.
[7] R. Fleischer. On the Bahncard problem. *Theoretical Computer Science*, 268(1):161–174, 2001.
[8] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP- Completeness*. W.H. Freeman, 1979.
[9] 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.
[10] A. R. Karlin, M. S. Manasse, L. Rudolph, and D. Sleator. Competitive snoopy caching. *Algorithmica*, 3(1):70–119, 1988.
[11] E. Nygren, R. K. Sitaraman, and J. Sun. The Akamai Network: A Platform for High-Performance Internet Applications. *ACM SIGOPS Operating Systems Review*, 44(3), July 2010.
[12] D. D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Commun. ACM*, 28(2):202–208, 1985.
[13] A. Borodin and R. El-Yaniv. Online Computation and Competitive Analysis. *Cambridge University Press*, 1998.
[14] D.K. Goldenberg, L. Qiu, H. Xie, Y. R. Yang, and Y. Zhang. Optimizing Cost and Performance for Multihoming. *ACM SIGCOMM*, Portland, Oregon, 2004.
[15] M. Adler, R. Sitaraman, H. Venkataramani. Algorithms for Optimizing Bandwidth Costs on the Internet. 1st IEEE Workshop on Hot Topics in Web Systems and Technologies (HOTWEB), pp.1-9, 2006.
[16] Iljitsch Van Beijnum, BGP. O'Reilly Media, 2002.
[17] R. Buyya, M. Pathan, A. Vakali (Eds.) Content Delivery Networks, Springer-Verlag, Germany, 2008.
[18] Pallis, G. and Vakali,. A. Insight and perspectives for content delivery networks, Communications of the ACM, 49(1), ACM Press, NY, USA, pp. 101-106, Jan. 2006.
[19] Gnutella. http://www.gnutella.com.
[20] KaZaa. http://www.kazaa.com.
[21] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. ACM SIGCOMM Computer Communication Review, Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, Volume 31 Issue 4, August 2001.
[22] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, Scott Schenker. A scalable content-addressable network. ACM SIGCOMM Computer Communication Review, Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, Volume 31 Issue 4, August 2001.
[23] Ben Y. Zhao, Ling Huang, Jeremy Stribling, Sean C. Rhea, Anthony D. Joseph, and John Kubiatowicz. Tapestry: A Resilient Global-scale Overlay for Service Deployment. IEEE Journal on Selected Areas in Communications, January 2004, Vol. 22, No. 1.
[24] A. Rowstron and P. Druschel. "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems". IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, pages 329-350, November, 2001.
[25] R. Buyya, A. M. K. Pathan, J. Broberg, and Z. Tari, A Case for Peering of Content Delivery Networks. IEEE Distributed Systems Online, Vol. 7, No. 10, IEEE CS Press, Los Alamitos, CA, USA, October 2006.