Designing Overlay Multicast Networks For Streaming

Konstantin Andreev* Bruce M. Maggs† Adam Meyerson‡ Ramesh K. Sitaraman§

ABSTRACT

In this paper we present a polynomial time approximation algorithm for designing a multicast overlay network. The algorithm finds a solution that satisfies capacity and reliability constraints to within a constant factor of optimal, and cost to within a logarithmic factor. The class of networks that our algorithm applies to includes the one used by Akamai Technologies to deliver live media streams over the Internet. In particular, we analyze networks consisting of three stages of nodes. The nodes in the first stage are the sources where live streams originate. A source forwards each of its streams to one or more nodes in the second stage, which are called reflectors. A reflector can split an incoming stream into multiple identical outgoing streams, which are then sent on to nodes in the third and final stage, which are called the sinks. As the packets in a stream travel from one stage to the next, some of them may be lost. The job of a sink is to combine the packets from multiple instances of the same stream (by reordering packets and discarding duplicates) to form a single instance of the stream with minimal loss. We assume that the loss rate between any pair of nodes in the network is known, and that losses between different pairs are independent, but discuss extensions in which some losses may be correlated.

*Mathematics Department, Carnegie-Mellon University, Pittsburgh PA 15213. Email: konst@cmu.edu

†Computer Science Department, Carnegie-Mellon University, Pittsburgh PA 15213. Email: bmm@cs.cmu.edu

‡Aladdin Project, Carnegie-Mellon University, Pittsburgh PA 15213. Research supported by NSF grant CCR-0122581. Email: adam@cs.cmu.edu

§Akamai Technologies Inc., 8 Cambridge Center, Cambridge MA 02142, and Department of Computer Science, University of Massachusetts, Amherst MA 01003. Research supported by NSF Career award No. CCR-97-03017. Email: ramesh@cs.umass.edu

SPAA'03, June 7–9, 2003, San Diego, California, USA.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed Applications*

General Terms

Algorithms, Design, Reliability, Theory.

Keywords

Network Design, Streaming Media, Approximation Algorithms, Network Reliability

1. INTRODUCTION

One of the most appealing applications of the Internet is the delivery of high-quality live audio and video streams to the desktop at low cost. Live streaming is becoming increasingly popular, as more and more enterprises want to stream on the Internet to reach a world-wide audience. Common examples include radio and television broadcast, events with a world-wide viewership, sporting events, and investor relation calls.

The traditional centralized approach to delivering live streaming involves three steps. First, the event is captured and encoded using an *encoder*. Next, the encoder delivers the encoded data to one more *media servers* housed in a centralized co-location facility on the Internet. Then, the media server streams the data to a media player on the end-user's computer. Significant advances in encoding technology, such as MPEG-2, have made it possible to achieve full-screen television quality video with data rates between 2 to 20 megabits per second. However, transporting the streaming bits across the Internet from the encoder to the end-user without significant loss in stream quality remains the critical problem, and is the topic of this paper.

The traditional centralized approach for stream delivery outlined above has two bottlenecks, both of which argue for the construction of an overlay distribution network for delivering live streams.

Server bottleneck. Most commercial media servers can serve no more than 50 Mbps of streams to end-users. In January 2002, Akamai hosted Steve Jobs's Keynote address at MacWorld-West which drew 50,000 simultaneous viewers world-wide with a peak traffic of 16.5 Gbps. To host an event of this magnitude, requires hundreds of servers. In addition these servers must be distributed across several co-location centers, since few co-location centers can provide even a tenth of the outgoing bandwidth required. Furthermore, a single co-location center is a single point of failure. Therefore, scalability and reliability requirements dictate the need for a distributed infrastructure consisting of a large number of servers deployed across the Internet.

Network bottleneck. As live events are increasingly streamed to a global viewership, streaming data needs to be transported reliably and in real-time from the encoder to the end-user's media player over the long haul across the Internet. The Internet is designed as a best-effort network with no quality guarantees for communication between two end points, and packets can be lost or delayed as they pass through congested routers or links. This can cause the stream to degrade, producing "glitches", "slide-shows", and "freeze ups" as the user watches the stream. In addition to degradations caused by packet loss, catastrophic events occasionally bring complete denial of service to segments of the audience. These events include complete failure of large ISP's, or failing of ISP's to peer with each other. As an example of the former on 10/3/2002, the WorldCom network experienced a total outage for nine hours. As an example of the latter, in June 2001, Cable and Wireless abruply stopped peering with PSINet for financial reasons. In the traditional centralized delivery model, it is customary to ensure that the encoder is able to communicate well with the media servers through a dedicated leased line, a satellite uplink, or through co-location. However, delivery of bits from the media servers to the end-user over the long haul is left to the vagaries of the Internet.

1.1 An overlay network for delivering live streams

The purpose of an overlay network is to transport bits from the encoder to the end-user in a manner that alleviates the server and network bottlenecks. The overlay network studied in this paper consists of three types of components, each globally distributed across the internet: entrypoints (also called sources), reflectors, and edgeservers (also called sinks), as shown in Figure 1. We illustrate the functionality of the three components by tracking the path of a stream through the overlay network as it travels from the encoder to the end-user's media player.

- An *entrypoint* serves as the point of entry for the stream into the overlay network, and receives the sequence of packets that constitutes the stream from the encoder. The entrypoint then sends identical copies of the stream to one or more reflectors.
- A *reflector* serves as a "splitter" and can send each stream that it receives to one or more edge-servers.
- An *edgeserver* receives one or more identical copies of the stream, each from a different reflector, and "reconstructs" a cleaner copy of the stream, before sending it to the media player of the end-user. Specifically, if the k^{th} packet is missing in one copy of the stream, the edgeserver waits for that packet to arrive in one of the other identical copies of the stream and uses it to fill the "hole".

The architecture of the overlay network described above allows for distributing a stream from its entrypoint to a large number of edgeservers with the help of reflectors, thus alleviating the server bottleneck. The network bottleneck can be broken down into three parts. The first-mile bottleneck from the encoder to the entrypoint can be alleviated by choosing an entrypoint close to (or even co-located with) the encoding facility. The middle-mile bottleneck of transporting bits over the long-haul from the entrypoint to the edgeserver can be alleviated by building an overlay network that supports low loss and high reliability. This is the hardest bottleneck to overcome, and algorithms for designing such a network is the topic of this paper. The last-mile bottleneck from the edgeserver to the end-user can be alleviated to a degree by mapping end-users to edgeservers that are "closest" to them. And, with significant growth of broadband into the homes of end-users, the last-mile bottleneck is bound to become less significant in the future[1].

1.2 Considerations for overlay network design

An overlay network can be represented as a tripartite digraph $N = (V, E)$ as shown in Figure 1, where V is partitioned into the set of entrypoints, a.k.a. sources (S), reflectors (R), and edgeservers, a.k.a. sinks (D). In this framework, overlay network design can be viewed as a multicommodity flow problem, where each stream is a commodity that must be routed from the entrypoint, where it enters the network, to a subset of the edgeservers that are designated to serve that stream to end-users. We assume that the subset of the edgeservers which want a particular stream is an input into our algorithm and takes into account the expected viewership of the stream, i.e., a large event with predominantly European viewership should include a large number of edgeservers in Europe in its designated subset, so as to provide many proximal choices to the viewers. Note that a given edgeserver can and typically will be designated to serve a number of distinct streams.

Given a set of streams and their respective edgeserver destinations, an overlay network must be constructed to minimize *cost*, subject to *capacity*, *quality*, and *reliability* requirements outlined below.

Cost: The primary cost of operating an overlay network is the bandwidth costs of sending traffic over the network The entrypoints, reflectors, and edgeservers are located in co-location centers across the Internet, and to operate the network requires entering into contracts with each co-location center for bandwidth usage in and out of the facility. A typical bandwidth contract is based either on average bandwidth usage over 5 minute buckets for the month, or on the 95^{th} percentile peak traffic usage in 5 minute buckets for the month. Therefore, depending on the specifics of the contract and usage in the month so far, it is possible to estimate the cost (in dollars) of sending additional bits across each link in the network. The total cost of usage of all the links is the function that we would like to minimize.

Capacity: There are capacity constraints associated with each entrypoint, reflector, and edgeserver. Capacity is the maximum total bandwidth (in bits/sec) that the component is allowed to send. The capacity bound incorporates CPU, memory, and other resource limitations on the machine, and bandwidth limitations on the outbound traffic from the co-location facility. For instance, a reflector machine may be able to push at most 50 Mbps before becoming CPU-bound. In addition to resource limitations, one can also use capacities to clamp down traffic from certain locations and move traffic around the network to control costs.

[1]From April 2001 to April 2002, the number of high-speed, home-based internet users in the US grew at an incredible 58%, from 15.9 million to 25.2 million individuals.

Quality: The quality of the stream that an edgeserver delivers to an end-user is directly related to whether or not the edgeserver is able to reconstruct the stream without a significant fraction of lost packets. Consequently, we associate a loss threshold for each stream and edgeserver that specifies the maximum post-reconstruction loss allowed to guarantee good stream quality for end-users viewing the stream from that edgeserver. Note that packets that arrive very late or significantly out-of-order must also be considered effectively useless, as they cannot be utilized in real-time for stream playback.
Reliability: As mentioned earlier, catastrophic events on the Internet from time to time cause large segments of viewers to be denied service. To defend against this possibility, the network must be monitored and the overlay network recomputed very frequently to route around failures. In addition, one can place systematic constraints on how the overlay network is designed to provide greater fault-tolerance. An example of such a constraint is to require that multiple copies of a given stream sent from an encoder are always sent to reflectors located in different ISPs. This constraint would protect against the catastrophic failure or peering problems of any single ISP. We explore this in sections 6.4 and 6.5.

1.3 Packet loss model

In practice the packet loss on each link can be periodically estimated by proactively sending test packets to measure loss on that link. One can average these numbers and get an estimate of the probability of each packet on the link being lost. Thus we will assume that the algorithm receives as an input the probability of failure on each link, say p and every packet on that link can be lost with an average probability of p. Notice that we don't assume that loss of packets on individual links are uncorrelated, but we will assume that losses on different links are independent (however in the extensions, section 6.3 to 6.5, we consider a model in which some link losses are related). Therefore if we have the same packet sent on two consecutive links with probabilities of failure respectively p_1 and p_2 then the probability of losing the packet on this path is $p_1+p_2-p_1p_2$. Similarly if the failure probabilities of two edges coming to a node are p_1 and p_2 respectively then the loss probability of the package at this node is p_1p_2. Observe that these loss rules are the same as in the network reliability problem [30], but we also have costs on the edges and multiple commodities. Since our algorithm is reasonably fast it can be reruned as often as needed so that the overlay network adapts to changes in the link failure probabilities or costs.

1.4 Other Approaches

One of the oldest alternative approaches is called "multicast" [6]. The goal of multicast is to reduce the total bandwidth consumption required to send the same stream to a large number of hosts. Instead of sending all of the data directly from one server, a multicast tree is formed with a server at the root, routers at the internal nodes, and end users at the leaves. A router receives one copy of the stream from its parent and then forwards a copy to each of its children. The multicast tree is built automatically as players subscribe to the screen. The server does not keep track of which players have subscribed. It merely addresses all of the packets in the stream to a special multicast address, and the routers take care of forwarding the packets on to all of the players that are subscribing to that address. Support for multicast is providing at both the network and link layer. Special IP and hardware addresses have been allotted to multicast, and many commercial routers support the multicast protocols.

Unfortunately, few of the routers on major backbones are configured to participate in the multicast protocols, so as a practical matter it is not possible for a server to rely on multicast alone to deliver its streams. The "mbone" (multicast backbone) network was organized to address this problem [7]. Participants in mbone have installed routers that participate in the multicast protocols. In mbone, packets are sent between multicast routers using unicast "tunnels" through routers that do not participate in multicast.

A second problem with the multicast protocols is that trees are not very resilient to failures. In particular, if a node or link in a multicast tree fails, all of the leaves downstream of the failure lose access to the stream. While the multicast protocols do provide for automatic reconfiguration of the tree in response to a failure, end users will experience a disruption while reconfiguration takes place. Similarly, if an individual packet is lost at a node or link, all leaves downstream will see the same loss. To compound matters, the multicast protocols for building the tree, which rely on the underlying network routing protocols, do not attempt to minimize packet loss or maximize available bandwidth in the tree.

The commercial streaming software does not rely on multicast, but instead provides a new component called a reflector. A reflector receives one copy of a stream and then forwards multiple copies on to other reflectors or streaming servers. A distribution tree can be formed by using reflectors as internal nodes, except for the parents of the leaves, which are standart media servers. As before, the leaves are media players. The reason for the layer of servers at the bottom of the tree is that the commercial software requires each player to connect individually to a server. The servers, players, and reflectors can all be configured to pull their streams from alternate sources in the event of failure. This scheme, however, suffers from the same disruptions and downstream packet loss as the multicast tree approach.

Recently promising new approaches have been devolped. One of them is "End System Multicast"(ESM) [3]. In ESM, there is no distinction between clients, reflectors, and servers. Each host participating in the multicast may be called on to play any of these roles simultaneously in order to form a tree. ESM is a peer-to-peer streaming applications, as it allows multicast groups to be formed without any network support for routing protocols and without any other permanent infrastructure dedicated to supporting multicast. Another one is "Cooperative Networking" (CoopNet) [24]. CoopNet is a hybrid between a centralized system as described in our paper and a peer-to-peer system such as ESM.

1.5 Related work

Our approach falls into the general class of facility location problems. Here the goal is to place a set of facilities (reflectors) into a network so as to maximize the coverage of demand nodes (sinks) at minimum cost. This class of problems has numerous applications in operations research, databases, and computer networking. The first approximation algorithm for facility location problems was given by Hochbaum [12] and improved approximatio algorithms have been the subject of numerous papers including [27, 9, 4, 2, 16, 29, 15, 22].

Except for Hochbaum's result, the papers described above all assume that the weights between reflectors and sinks form a metric (satisfying the symmetry and triangle inequality properties). In our problem, the weights represent transmission failure probabilities. These probabilities do not necessarily form a metric. For example, the symmetry constraint frequently fails in real networks. Without the triangle inequality assumption, the problem is as hard as set cover, giving us an approximation lower bound of $O(\log n)$ with respect to cost for polynomial-time computation (unless $NP \subset DTIME(n^{O \log\log n}))$ [21, 8]. A simple greedy algorithm gives a matching upper bound for the set cover problem [18, 5].

While our problem includes set cover as a special case, the actual problem statement is more general. Our facilities are capacitated (in contrast to the set cover problem where the sets are uncapacitated). Capacitated facility location (with "hard" capacities) has been considered by [25], but the local search algorithm provided depends heavily upon the use of an underlying metric space. The standard greedy approach for the set cover problem can be extended to accommodate capacitated sets, but our problem additionally requires an assignment of both commodities to reflectors and reflectors to sinks. Similar two-level assignments have been considered previously [20, 1, 23, 11], but again the earlier work assumed that the points were located in a metric space. The greedy approach may not work for multiple commodities, as the coverage no longer increases concavely as reflectors are added. In other words, adding two reflectors may improve our solution by a larger margin than the sum of the improvements of the reflectors taken individually.

Our goal is to restrict the probability of failure at each node, and it will typically be necessary to provide each stream from more than one reflector. This distinguishes our problem from most previous work in set cover and facility location, where the goal is to cover each customer with exactly one reflector. Several earlier papers have considered the problem of facility location with redundancy [17, 10]. Unlike our results, each of the previous papers assumes an underlying metric, and it is also assumed that the coverage provided by each facility is equivalent (whereas in our problem the coverage provided is represented by the success rate and depends upon the reflector-customer pair in question).

The problem of constructing a fault-tolerant network has been considered previously. The problem is made difficult by dependencies in the failure rates. Selecting a set of paths to minimize the failure rate between a pair of nodes is made difficult by the fact that intersecting paths are not independent (but their combined probability of failure is still less than the failure probability of any path individually). Earlier papers have considered network reliability. For general networks Valiant [30] defined the term "network reliability" and proved that computing it is ♯P-complete. Karger showed an FPRAS that approximates the network reliability [19]. We consider a three-tiered network because these structures are used in practice (for example in Akamai's data-distribution network) and because the possible dependencies between paths are greatly reduced in such a network (two hop paths only recombine at the last level). In such a network one can compute the exact reliability in polynomial time. If we consider our problem as a sort of weighted capacitated set cover, it would be straightforward to extend the results to any network of constant depth. However, since the weights represent probabilities of failure, our results do not directly extend to constructing a reliable network with more than three layers (the chance of failure at a customer would no longer be equal to the product of failure probabilities along paths since the paths need not be independent in a deeper network).

1.6 Our results

Our techniques are based upon linear program rounding, combined with the generalized assignment algorithm of [26]. A direct rounding approach is possible, but would lead to a multicriterion logarithmic approximation. We are forced to lose $O(\log n)$ on the cost (due to the set cover lower bounds), but we obtain $O(1)$ approximation bounds on the capacity and probability requirements by using randomized rounding for only some linear program variables and completing the rounding procedure by using a modified version of generalized assignment. In Section 6 we use a technique due to Srinivasan and Teo [28] to tackle some extensions of this problem. The constants can be traded off in a manner typical for multicriterion approximations, allowing us to improve the constants on the capacity and probabilities by accepting a larger constant multiplier to the cost. Our algorithm is randomized, and the randomized rounding makes use of Chernoff bounds as extended by Hoeffding [13, 14].

1.7 Outline of the paper

The remainder of this paper is organized as follows. In Section 2 we formalize the problem. In Section 3 we describe the randomized rounding procedure which is the first stage of our algorithm. In Section 4 we analyze the effect that the rounding procedure has on the fractional solution of the LP. In Section 5 we describe the second stage of the algorithm - the modified generalized assignment problem approximation and analyze it. In Section 6 we suggest various extensions and generalizations of the problem and what we know about them. In Section 7 we talk about future directions.

2. PROBLEM DESCRIPTION

The *3-level network reliability min-cost multicommodity flow problem* is defined as follows: We are given sets of sources and destinations in a 3-partite digraph $N = (V, E)$ where $V = S\dot{\cup}R\dot{\cup}D$ with costs on the edges

$$c^u : \mathrm{E}^u \to \Re_+^u$$

where u is the number of commodities,i.e. the cost for carrying a commodity may vary, perhaps to capture different encoding ratios. Costs for building a node on the middle level (will call all the nodes in the middle level reflectors)

$$r : \mathrm{R} \to \Re_+$$

probabilities of failure on the edges

$$p : \mathrm{E} \to [0, 1]$$

and demand threshold for each destination and different commodity

$$\Phi^u : \mathrm{D}^u \to [0, 1]^u.$$

There are also fanout constraints F_i on each reflector $i \in R$. The problem is to find a minimum cost subnetwork such that when we send a packet, which is lost at each edge with some given probability, we are still assured that each sink will receive at least one copy of the packet with probability at least equal to the demand. The primary difference from previous network flow problems is that we don't have preservation of flow at each node. Instead if a flow is received at a reflector $i \in R$ it can be sent simultaneously to as many neighbors as its fanout F_i. The cost of routing along an arc may depend upon the commodity being sent. We describe an algorithm which approximates this min cost integer flow problem. This problem can model *SET COVER*. Thus the best solution in terms of cost that we can hope for, unless $NP \subset DTIME(n^{O(\log\log n)})$, is a $O(\log n)$ approximation [8]. Our problem is more general than set cover in several ways. We introduce fanout constraints on the reflectors (effectively, each set can cover only some of its elements). We also have costs, both on the reflectors themselves and on covering a sink with a reflector, and we require that each sink must be covered by multiple reflectors (typically single coverage is not enough) which ensure at least the required success probability. We present an LP rounding solution to the problem which has a guarantee of $O(\log n)$ approximation on the cost and violates the probability and fan out constraints by small constants.

Without loss of generality, we assume that each sink has a non-zero demand for only one commodity. We can do this by replacing each single sink by multiple copies. Once this modification is made, we let n denote the number of sinks, i.e. $n = |D|$. For simplicity of notation we further assume that each source sends its own commodity, therefore $|S| = u$.

We define the integer program (IP) that models the problem. We use y_i^k as the indicator variable for delivery of the k-th stream to the i-th reflector, z_i as the indicator variable for building reflector i and x_{ij}^k as the indicator variable for delivering the k-th stream to the j-th sink through the i-th reflector. F_i denotes the fanout constraint for each $i \in R$. We transform the probabilities into weights: $w_{ij}^k = -\log(p_{ki} + p_{ij} - p_{ki}p_{ij})$ for the probabilities on the edges. Here p_{ij} is the failure probability on edge ij and p_{ik} is the failure of commodity k reaching reflector i. In other words w_{ij}^k is the negative log of the (failure) probability that a commodity k, originating from source k fails to reach sink j. On the other hand $W_j^k = -\log(1 - \Phi_j^k)$ for the demand weight, where Φ_j^k is the minimum required success probability. That means W_j^k is the negative log of the maximum allowed failure. Thus we are able to write the IP:

$$\begin{array}{ll} \min & \sum_{i\in R} r_i z_i + \sum_{i\in R}\sum_{k\in S} c_{ki}^k y_i^k + \sum_{i\in R}\sum_{k\in S}\sum_{j\in D} c_{ij}^k x_{ij}^k \\ s.t. & \end{array}$$

(1) $\quad y_i^k \le z_i \;\; \forall i \in R,\, \forall k \in S$

(2) $\quad x_{ij}^k \le y_i^k \;\; \forall i \in R,\, \forall j \in D,\, \forall k \in S$

(3) $\quad \sum_{k\in S}\sum_{j\in D} x_{ij}^k \le F_i z_i \;\; \forall i \in R$

(4) $\quad \sum_{j\in D} x_{ij}^k \le F_i y_i^k \;\; \forall i \in R,\, \forall k \in S$

(5) $\quad \sum_{i\in R} x_{ij}^k w_{ij}^k \ge W_j^k \;\; \forall j \in D,\, \forall k \in S$

(6) $\quad x_{ij}^k \in \{0,1\},\; y_i^k \in \{0,1\},\; z_i \in \{0,1\}$

Constraint (1) and (2) force us to pay for the reflectors we are using, and to transmit packets only through reflectors which are in use. Constraint (3) encodes the fanout restriction. Constraint (4) is redundant in the IP formulation, but provides a useful cutting plane in the rounding. Constraint (5) is the reliability condition, requiring that we obtain sufficient weight at each sink. Constraint (6) is integrality, and will be relaxed in the LP formulation.

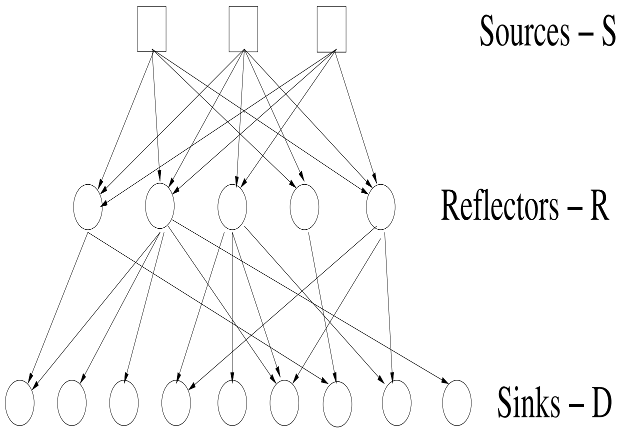

Figure 1: 3-level Network

Claim 2.1. *In the IP formulation constraints (1),(2),(3) and (6) dominate (4).*

Proof. We look at cases for z_i.

1) If $z_i = 0$, then from (1) and (6) we get $y_i^k = 0$ for $\forall k \in S$. Now from (2) and (6) we get $x_{ij}^k = 0 \;\; \forall k \in S$ and $\forall j \in D$. Thus (4) is implied.

2) If $z_i = 1$, then if $y_i^k = 0$ we still have $x_{ij}^k = 0 \;\; \forall j \in D$, which means

$$\sum_{j\in D} x_{ij}^k = 0$$

If $y_i^k = 1$ then from (3) we have

$$\sum_{k\in S}\sum_{j\in D} x_{ij}^k \le F_i$$

which means that $\forall k \in S$

$$\sum_{j\in D} x_{ij}^k \le F_i$$

Which concludes the proof. $\square$

We will find an approximate solution to the above IP using randomized rounding. We know that the corresponding LP relaxation is obtained by just substituting the integrality constraints (6) in the IP with

$$x_{ij}^k \in [0,1],\; y_i^k \in [0,1],\; z_i \in [0,1]$$

We solve the LP to optimality and find a fractional solution

$$(\hat{z}_i,\;\; \hat{y}_i^k,\;\; \hat{x}_{ij}^k)$$

3. RANDOMIZED ROUNDING

We will use parameter $c > 1$, which will be determined later, as a preset multiplier. We apply the following randomized rounding procedure, where by $(\bar{z}_i,\ \bar{y}_i^k,\ \bar{x}_{ij}^k)$ we denote the rounded values

[1] Compute $\dot{z}_i = \min(\hat{z}_i c \log n, 1)\ \ \forall i \in R$

[2] Compute $\forall i \in R$, $\forall k \in S$

$$\dot{y}_i^k = \min\left(\frac{\hat{y}_i^k c \log n}{\dot{z}_i}, 1\right)$$

[3] We round $\bar{z}_i = 1$ with probability $\dot{z}_i$ and 0 otherwise.

[4] If $\bar{z}_i = 1$ then round $\bar{y}_i^k = 1$ with probability $\dot{y}_i^k$ and 0 otherwise.

[5] If $\dot{z}_i = \dot{y}_i^k = 1$ set $\bar{x}_{ij}^k = \hat{x}_{ij}^k$
else if $\bar{y}_i^k = 1$ set $\bar{x}_{ij}^k = \frac{1}{c \log n}$ with probability $\hat{x}_{ij}^k / \hat{y}_i^k$.

[6] Set all the other variables to 0.

The only fractional values left after this procedure are $\bar{x}_{ij}^k$. To round them we will apply a modified version of the Generalized Assignment Problem (GAP) approximation due to Shmoys and Tardos [26]. It will preserve the cost and violate the fan out and weight constraints by at most a constant factor.

4. ANALYSIS OF THE RANDOMIZED ROUNDING

Let $\hat{C}$ denote the value of the objective function for our fractional solution, with $\bar{C}$ the value after the rounding procedure, and with C^{OPT} the optimal IP value. From the rounding procedure it's clear that $\mathbf{E}[\bar{z}_i] \leq \hat{z}_i c \log n$, $\mathbf{E}[\bar{y}_i^k] \leq \hat{y}_i^k c \log n$ and $\mathbf{E}[\bar{x}_{ij}^k] = \hat{x}_{ij}^k$. These three inequalities imply

$$\mathbf{E}[\bar{C}] \leq c \log n \cdot \hat{C} \leq c \log n \cdot C^{OPT}.$$

Thus we have the following lemma.

Lemma 4.1. *The expected cost after the rounding is at most $c \log n$ times the optimal cost.*

Now we will show that with high probability the weight constraints are violated by a small constant factor and the fan out constraints - by at most a factor of two. Combining this with the GAP approximation will yield a solution to the IP which has a cost at most $c \log n$ times optimal and violates the fan out and weight constraints by at most a factor of 4. By high probability, we mean a probability of less than $1/n$ of violating any of the constraints. First we will look at the weight constraint, i.e.

$$\sum_{i \in R} x_{ij}^k w_{ij}^k \geq W_j^k\ \ \forall j \in D,\ \forall k \in S$$

From the rounding procedure it is clear that some of the $\bar{x}_{ij}^k$ are deterministic. We will decompose those numbers and think of them as random variables equal to $\frac{1}{c \log n}$ with probability 1. We define random variable $v_i = (c \log n)\bar{x}_{ij}^k \frac{w_{ij}^k}{W_j^k}$ (notice that j and k are fixed). Without loss of generality we can assume $w_{ij}^k \leq W_j^k$ since it never helps to have more weight on an edge than the one that a sink demands. Therefore $v_i \in [0,1]$. Also let's note that v_i are all independent since they depend on different $y_i^k$for every i. The expected value of v_i is

$$\mathbf{E}[v_i] = c \log n \cdot \frac{w_{ij}^k}{W_j^k} \cdot \hat{x}_{ij}^k$$

We are going to use a generalized version of the *Chernoff bound* where the random variables $v_i \in [0,1]$.

Theorem 4.2 (Hoeffding-Chernoff bound). *For $v_i \in [0,1]$ independent random variables, let $S = \sum_i v_i$ and $\mu = \mathbf{E}\left[\sum_i v_i\right]$ then*

$$\Pr(S \leq (1-\delta)\mu) \leq exp\left(-\tfrac{\delta^2 \mu}{2}\right)$$

$$\Pr(S \geq (1+\delta)\mu) \leq exp\left(-\tfrac{\delta^2 \mu}{3}\right)$$

Proof is defered to *Appendix A*. We observe that

$$\mu = \mathbf{E}\left[\sum_{i \in R} v_i\right] = \sum_{i \in R} \mathbf{E}[v_i] = c \log n \cdot \sum_{i \in R} \frac{w_{ij}^k}{W_j^k} \cdot \hat{x}_{ij}^k \geq c \log n.$$

Lets denote with $\overline{W}_j^k$ the sum of weights over $i \in R$ after the rounding step. Using the *Hoeffding-Chernoff* theorem, we get the following chain of inequalities

$$\Pr(\overline{W}_j^k < (1-\delta) W_j^k) = \Pr\left(\textstyle\sum_{i \in R} w_{ij}^k \cdot \bar{x}_{ij}^k < (1-\delta) W_j^k\right) \leq$$

$$\leq \Pr\left(\textstyle\sum_{i \in R} \frac{w_{ij}^k \cdot \bar{x}_{ij}^k}{W_j^k} < (1-\delta) \sum_{i \in R} \frac{w_{ij}^k \cdot \hat{x}_{ij}^k}{W_j^k}\right) =$$

$$= \Pr(\textstyle\sum_{i \in R} v_i \leq (1-\delta)\mu) \leq exp\left(-\frac{\delta^2 \mu}{2}\right).$$

Which implies that the probability of a particular weight constraint (one for a fixed j and k) to be violated by a factor of $1/(1-\delta)$ is

$$\Pr\left(\overline{W}_j^k \cdot \bar{x}_{ij}^k < (1-\delta) W_j^k\right) \leq e^{(-\frac{\delta^2 \cdot c \log n}{2})} = \frac{1}{n^{\delta^2 \cdot c/2}}.$$

Here we get a trade-off between a tighter constant with which we violate the weight inequalities and the competitive cost ratio against an integral optimal solution. As we said, our goal is to achieve a probability of violating any of the constraints less than $1/n$. Since there are exactly n weight constraints we need to set $\delta^2 \cdot c = 4$. If $\delta = 1/4$ then $c = 64$. We summarize these results as follows:

Lemma 4.3. *After the rounding procedure with high probability each of the weight constraints will be violated by at most a small constant factor.*

Now we look at the fan out constraints. As noted before the only set of fan out constraints needed in the IP is the following

$$\sum_{k \in S} \sum_{j \in D} x_{ij}^k \leq F_i z_i\ \ \forall i \in R.$$

We want to again apply the *Hoeffding-Chernoff bound.* Unfortunately from the rounding procedure it's clear that knowing $\bar{y}_i^k = 1$ gives higher probability for $\forall j \in D$ that $\bar{x}_{ij}^k$ are rounded to $1/c \log n$. In other words $\bar{x}_{ij}^k$ are no longer independent random variables. However $\bar{x}_{ij}^k$ are obtained by a two stage process in which first $\bar{y}_i^k$ is rounded to 0 or 1 and then $\bar{x}_{ij}^k$ is rounded iff $\bar{y}_i^k = 1$. We will use two claims to proof the next lemma.

CLAIM 4.4. *For a probability space over the $\bar{y}_i^k$ we have*

$$\Pr\left(\mathbf{E}\left[\sum_{k\in S}\sum_{j\in D}\bar{x}_{ij}^k|\bar{y}_i^k\right] > \frac{3}{2}F_i\right) < \frac{1}{2n^2}$$

PROOF. We use linearity of expectation to get

$$\mathbf{E}\left[\sum_{k\in S}\sum_{j\in D}\bar{x}_{ij}^k|\bar{y}_i^k\right] = \sum_{k\in S}\mathbf{E}\left[\sum_{j\in D}\bar{x}_{ij}^k|\bar{y}_i^k\right]$$

Let's look at cases for a particular $\bar{y}_i^k$. Either $\bar{y}_i^k = 0$ then

$$\mathbf{E}\left[\sum_{j\in D}\bar{x}_{ij}^k|\bar{y}_i^k\right] = 0$$

Or $\bar{y}_i^k = 1$ then from the cutting plane equation (4) we have

$$\mathbf{E}\left[\sum_{j\in D}\bar{x}_{ij}^k|\bar{y}_i^k\right] = \sum_{j\in D}\frac{1}{c\log n}\cdot\frac{\hat{x}_{ij}^k}{\hat{y}_i^k} \le \frac{F_i}{c\log n}$$

We know from equation (3) that

$$\mathbf{E}\left[\sum_{k\in S}\sum_{j\in D}\bar{x}_{ij}^k\right] \le F_i$$

Now we use the *Hoeffding-Chernoff bound* and setting $c \ge 24$ we get

$$\Pr\left(\mathbf{E}\left[\sum_{k\in S}\sum_{j\in D}\bar{x}_{ij}^k|\bar{y}_i^k\right] > \frac{3}{2}F_i\right) < \frac{1}{2n^2}$$

Which concludes the proof of this claim. $\square$

The second claim is

CLAIM 4.5. *Suppose that for some fixed $\bar{y}_i^k$ that*

$$\mathbf{E}\left[\sum_{k\in S}\sum_{j\in D}\bar{x}_{ij}^k|\bar{y}_i^k\right] \le \frac{3}{2}F_i$$

Then for $c \ge 24$

$$\Pr\left(\sum_{k\in S}\sum_{j\in D}\bar{x}_{ij}^k > 2F_i\right) < \frac{1}{2n^2}$$

PROOF. When all $\bar{y}_i^k$ are fixed then $\bar{x}_{ij}^k$ are independent. Thus we apply straight forward *Hoeffding-Chernoff bound* and we get the bound. $\square$

We summarize these results in the following

LEMMA 4.6. *If we set $c \ge 24$ then after the rounding procedure with high probability each of the fan out constraints will be violated by at most a factor of 2.*

5. ROUNDING BY MODIFIED GAP APPROXIMATION

As the last part of the approximation algorithm we will describe how to convert the $\bar{x}_{ij}^k$ after the rounding procedure to an integral solution. This solution will violate the fan out constraints by an additional factor of two, for a combined factor of 4 and will violate the weight constraint by a combined factor of 4. As before let us denote with $\bar{C}$ the cost achieved by $\bar{x}_{ij}^k$. We design the following five level network. We start with a source s that is connected to each reflector i in the second level of all reflectors with an edge of capacity equal to the fan out of the reflector, F_i. For each reflector i in the third level we list its sinks with $\bar{x}_{ij}^k \ne 0$ and put an edge of capacity 1. That is the third level consists of nodes representing (reflector, sink) pairs such that $\bar{x}_{ij}^k \ne 0$ for at least one k. In the fourth level we represent each sink as a collection of boxes where the number of boxes is equal to

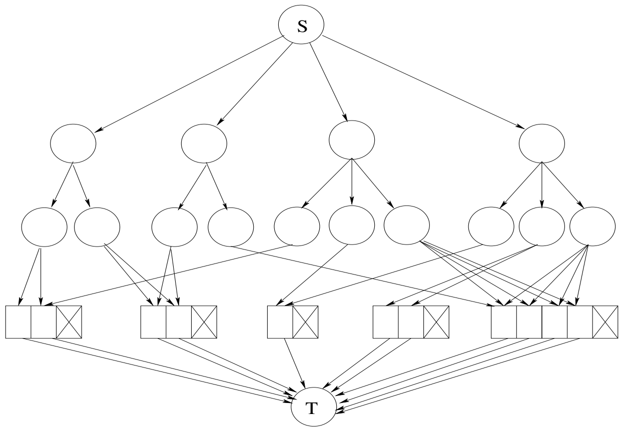

Figure 2: $\bar{x}_{ij}^k$ fractional solution conversion network

$$s_j = \left\lceil 2\sum_{i\in R}\bar{x}_{ij}^k\right\rceil.$$

We order the w_{ij}^k for each sink in decreasing order, WLOG

$$w_{1j}^k \ge w_{2j}^k \ge \ldots$$

This gives us an ordering on the nonzero $\bar{x}_{ij}^k$ Then with each box we associate an interval of weights. Let s be the first index for which

$$\sum_{i=1}^{s}\hat{x}_{ij}^k > \frac{1}{2}.$$

Then the first box will have the interval $[w_{1j}^k, w_{sj}^k]$ associated with it. We set $x' = \sum_{i=1}^{s}\hat{x}_{ij}^k - 1/2$. If $x' > 1/2$ we have $r = s$ and we mark the box with $[w_{rj}^k, w_{rj}^k]$.Otherwise we look for the index for which

$$x' + \sum_{i=s+1}^{r}\bar{x}_{ij}^k > \frac{1}{2}.$$

and we mark the second box with $[w_{sj}^k, w_{rj}^k]$. Continue with this algorithm until we fill all the boxes except possibly the last one. We then eliminate the last box for each sink. Then we connect each (reflector, sink) pair from level 3 to some of its corresponding sink boxes on level 4. More precisely whenever the corresponding w_{ij}^k is in the interval range associated with the box on level 4 for the sink we place an edge of capacity 1/2 between the pair and the box. Finally we connect all the boxes to a sink T with edges of capacity 1/2. The demand is, then is equal to the sum of 1/2 over all edges from level 4 to the sink T. From the construction it is clear that the fractional flow $\bar{x}^k_{ij}$, reduced so as to obey the edge capacities, saturates the demand at the sink T. Thus there exists a maximum flow with flow variables equal to 0, 1/2 or 1 that has a cost at most $\bar{C}$. If we assume $c \geq 64$ then we know that $\overline{W}^k_j \geq \frac{3}{4}W^k_j$. Thus for any flow we will have weight at least:

$$\tfrac{1}{2}\textstyle\sum_{\ell=1}^{s_j-1}\min(w^k_{\ell j}) \geq \geq \tfrac{1}{2}\sum_{\ell=2}^{s_j}\max(w^k_{\ell j})$$

$$\geq \textstyle\sum_{i\in R} w^k_{ij}\bar{x}^k_{ij} - \tfrac{1}{2}w^k_{1j} \geq \overline{W}^k_j - \tfrac{1}{2}W^k_j \geq \tfrac{1}{4}W^k_j.$$

Here by max or min we mean the upper or lower bound of the interval ℓ. So the resulting flow satisfies at least half the weight demand of each sink. Now we double all $x^k_{ij} = 1/2$. Thus we might have violated each of the weight and fan out constraints by at most a factor of two. We also double the cost associated with x^k_{ij} but that is already accounted for since we have an $O(\log n)$ factor on the cost because of the rounding of $\hat{y}^k_i$ and $\hat{z}_i$. This concludes the rounding of the last fractional variables of our solution. We get a 0-1 solution.

Here is some intuition of what a 4-approximation guarantee on the weight means in our context. Since we started by converting probabilities into weights using log, a factor of 4 violation translates into 4-th root of the failure probabilities. For example if we want success of $\Phi^k_i = .9999$ that is failure of less than .0001 what we have is a .9 guarantee or a failure probability of at most .1.

5.1 Running Time

We will conclude this section by calculating the running time of our approximation algorithm. Observe that the initial LP has $O(|S|\cdot|R|\cdot|D|)$ variables and constraints. Here S is the number of streams and D is the number of (stream, sink) pairs when a sink wants to view a stream. The LP rounding step takes as many iterations as the number of LP variables, so we can include it's running time in the LP solver step. The modified GAP network has $O(|R|\cdot|D|)$ nodes and edges. The running time of solving the network flow problem is absorbed by the LP solver step. Therefore the total running time of our algorithm is the same as solving an LP with $O(|S|\cdot|R|\cdot|D|)$ variables and constraints.

6. EXTENSIONS

In this section we examine several extensions and generalizations of the problem.

6.1 Bandwidth on reflectors

Let's put capacities on the ability of each reflector to route different flow. We consider the following modification to constraints (3) and (4):

$$(3') \qquad \textstyle\sum_{k\in S} B^k \cdot \sum_{j\in D} x^k_{ij} \leq F_i z_i \;\; \forall i \in R$$

$$(4') \qquad B^k \cdot \textstyle\sum_{j\in D} x^k_{ij} \leq F_i y^k_i \;\; \forall i \in R,\ \forall k \in S$$

Here $B^k \in \Re_+$ can be viewed as a bandwidth for each stream that enters a reflector. Now with small modifications the whole analysis goes through. This allows us to model the service by reflectors of different bandwidth streams.

6.2 Capacities on all of the arcs

Now we consider a capacitated version of the problem, i.e. we add new constraints

$$(7) \qquad \textstyle\sum_{k\in S} x^k_{ij} \leq u_{ij} \;\; \forall i \in R, \forall j \in D$$

$$(8) \qquad \textstyle\sum_{k\in S} y^k_i \leq u_i \;\; \forall i \in R$$

Here

$$u : \mathrm{E} \rightarrow \Re_+.$$

If we assume that there exists a randomized algorithm which solves this modification of the problem by violating constraints (7) and (8) with a constant factor, then we showed there will be an algorithm that approximates *Set cover* to with in a constant factor. Since the latter is highly unlikely [8] there is not much hope for an interesting solution to this version of the problem. Note that our rounding procedure described before, applied to a fractional solution of the LP relaxation of the modified problem, will yield a $c\log n$ factor violation of constraints (7) and (8) - the best guarantee we can hope for.

6.3 Capacities between reflectors and sinks

We consider constraints which represent capacities between reflectors and sinks.

$$(7') \qquad \sum_{k\in S} x^k_{ij} \leq u_{ij} \;\; \forall i \in R, \forall j \in D.$$

Here

$$u : \mathrm{E(R,D)} \rightarrow \Re_+.$$

and $E(R, D)$ are all edges between reflector nodes and sinks.

6.4 Color constraints

We introduced another set of constraints, called color constraints. First let $R = R_1 \dot{\cup} R_2 \dot{\cup} R_3 \ldots \dot{\cup} R_m$. We have the following constraints added to the (IP)

$$(9) \qquad \sum_{i\in R_\ell} x^k_{ij} \leq 1. \;\; \forall j \in D,\ \forall k \in S,\ \forall \ell \in [m].$$

The idea behind these constraints is to break the reflectors into disjoint groups. Then we want to make sure that no group is delivering more than one copy of the stream into a sink. In terms of real life networks we can think of the groups as reflectors belonging to the same ISP. Thus we want to make sure that a client is served only with one, the best (or sufficient), stream possible from a certain ISP and thus diversifying the stream distribution over different ISPs. The advantage here is some stability in the solution - if one of the ISPs goes down we will still serve most of the sinks.

6.5 Solution

In this subsection we describe the solution to the last two extensions. We were able to solve them both using the same method. Of course introducing the 6.3 Capacity between reflectors and sinks and 6.4 Color constraints in the LP rounding is straight forward. It does not affect the rounding procedure from Section 3). The final step of rounding $\bar{x}^k_{ij}$ by modified GAP requires modification. Both extensions introduce a new type of constraint in the modified GAP network (Figure 2). This constraint bounds the total flow along some subsets of the edges between the second and third level of the GAP network. Such constraints can be introduced into

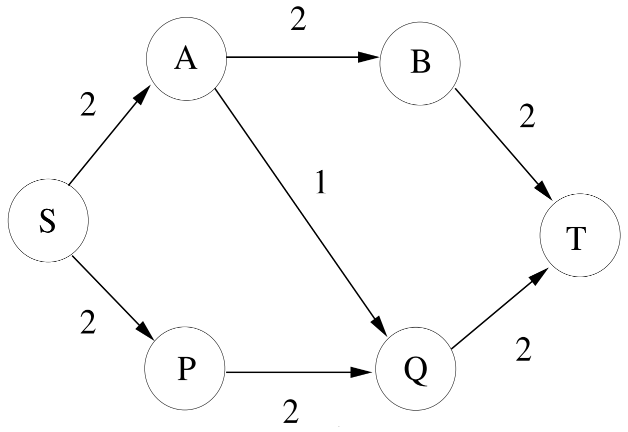

Figure 3: The capacities for all edges are shown in the figure. There is an additional set constraint that the set of edges $\{ab, pq\}$ has a capacity of 3

any flow type problem. As shown by the simple example in the figure, the introduction of such constraint creates a gap between the optimal fractional and integral flows. Clearly the max integral flow is only 3. However one can achieve a fractional max flow of 3.5 units, by sending 2 units of flow on sa and 1.5 units on edge sp then spliting the flow at a by sending .5 units on edge aq and the rest on ab. This will prevent us from applying GAP directly, as we cannot find an integral flow which is at least as good as the fractional flow.

Our approach finds an integral solution within a constant factor (less than 14) of optimal cost while violating the constrants by an additional constant factor (less than 7) by applying the techniques of Srinivasan and Teo [28]. We reformulate the network LP from section 5 in terms of paths. Let $\mathcal{P}$ be the set of all paths in Figure 2 from s to the boxes on level 4, $\mathcal{B}$ be the set of boxes (nodes) at level 4 and S_i be all sets of entangled edges. Notice that all S_i contain only edges between levels 2 and 3. We use the variable y_p to indicate whether path p is used to carry a flow, for each $p \in \mathcal{P}$. Here is the LP formulation:

$$
\begin{array}{lll}
(i) & \displaystyle\sum_{p\in\mathcal{P}|e\in p} y_p \leq 4u_e & \forall\, e \in E \\
(ii) & \displaystyle\sum_{p\in\mathcal{P}|p=\{s\to b\}} y_p = 1 & \forall\, b \in \mathcal{B} \\
(iii) & \displaystyle\sum_{p\in\mathcal{P}|p\cap S_i\neq\emptyset} y_p \leq 4u_i & \forall\, i \in [m] \\
(iv) & \displaystyle\sum_{p\in\mathcal{P}} c_p y_p \leq 2X &
\end{array}
$$

Here u_e is the capacity on edge $e \in E$, s is the source, $\{s \to b\}$ denotes a path from s to a box b, u_i is the capacity of set S_i, c_p is the cost of path $p \in \mathcal{P}$. X is the total cost of the solution produced by the randomized rounding stage. The first constraints (i) are capacities on the edges. Constraints (ii) require a flow of half to each of the boxes at the bottom layer. Constraints (iii) are the special set type constraints and constraint (iv) controls the cost. We can produce a feasible fractional solution to this program by taking our solution after the first stage of LP rounding and doubling all the flows. If this linear program was simply a single-commodity flow (i.e. without the constraints of type (iii)) then we could immediately transform our fractional solution to an integral solution.

Instead, we must apply Srinivasan and Teo's technique. We first upper and lower bound the positive and negative coefficients in front of any y_p. However since the cost can be arbitrary we need to make two modifications to the last constraint. We will first eliminate any paths with $c_p > 4X$ and then divide both sides of the inequality by $2X$. The eliminated paths are more than twice as expensive than the whole optimal solution and dropping them introduces at most a factor of 2 in the cost. Counting, we now have: y_p appears 4 times (at most once for each level) in (i), at most once in (iii) and exactly once in (iv) with coefficient less than 2. This adds up to a total of 7. Thus we multiply (ii) by negative 7. Applying Theorem 2.2 from Srinivasan and Teo we get an integral solution which satisfies all the constraints with an additive factor of 7. This factor translates into multiplicative factor of 14 for the cost. Thus we get the promised approximation guarantees.

The running time of this step is dominated by applying Theorem 2.2 from Srinivasan and Teo [28]. The number of non zero y_p variables, which corresponds to the r parameter in Srinivasan and Teo [28], is $O(|R| \cdot |D|)$. Thus the running time is at most $O(|R|^3 \cdot |D|^3)$.

7. FUTURE WORK

We also plan to implement the algorithm described in this paper (or heuristics based on the algorithm) and apply them to real-world network data gleaned from Akamai's streaming network.

8. REFERENCES

[1] I. Baev and R. Rajaraman. Approximation algorithms for data placement in arbitrary networks. *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms*, 2001.

[2] M. Charikar and S. Guha. Improved combinatorial algorithms for facility location and k-median problems. *Proceedings of 40th IEEE FOCS*, 1999.

[3] Y. Chu, S. Rao, and H. Zhang. A case for end system multicast. *Proceedings of ACM SIGMETRICS*, 2000.

[4] F. Chudak. Improved algorithms for uncapacitated facility location problem. *Proceedings of the 6th Conference on Integer Programming and Combinatorial Optimization*, 1998.

[5] V. Chvatal. A greedy heuristic for the set covering problem. *Math. Operations Research*, 1979.

[6] S. Deering. Multicast routing in a datagram internetwork. *Ph. D. Dissertation*, 1991.

[7] H. Eriksson. Mbone: The multicast backbone. *Communications of the ACM, 37(8)*, 1994.

[8] U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 1998.

[9] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[10] S. Guha, A. Meyerson, and K. Munagala. Improved algorithms for fault-tolerant facility location. *Proceedings of the 12th Annual ACM-SIAM SODA*, 2001.

[11] S. Guha and K. Munagala. Improved algorithms for the data placement problem. *Proceedings of the 13th ACM-SIAM Symposium on Discrete Algorithms*, 2002.
[12] D. Hochbaum. Heuristics for the fixed cost median problem. *Mathematical Programming*, 1982.
[13] W. Hoeffding. Probability inequalities for sums. *American Statistical Assocation Journal*, 1963.
[14] M. Hofri. *Probabilistic Analysis of Algorithms.* Springer–Verlag, 1987.
[15] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problem. *Proceedings of the 34th ACM STOC*, 2002.
[16] K. Jain and V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. *Proceedings of 40th IEEE FOCS*, 1999.
[17] K. Jain and V. Vazirani. An approximation algorithm for the fault tolerant metric facility locaiton problem. *APPROX*, 2000.
[18] D. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 1974.
[19] D. Karger. A randomized fully polynomial time approximation scheme for all-terminal network reliability problem. *Proceedings of the 27th ACM STOC*, 1995.
[20] M. Korupolu, G. Plaxton, and R. Rajaraman. Placement algorithms for hierarchical cooperative caching. *Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms*, 1999.
[21] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 1994.
[22] M. Mahdian, Y. Ye, and J. Zhang. Improved approximation algorithms for metric facility location problems. *APPROX*, 2002.
[23] A. Meyerson, K. Munagala, and S. Plotkin. Web caching using access statistics. *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms*, 2001.
[24] V. Padmanabhan, H. Wang, P. Chou, and K. Sripanidkulchai. Distributing streaming media content using cooperative networking. *To appear in ACM NOSSDAV*, 2002.
[25] M. Pál, E. Tardos, and T. Wexler. Facility location with nonuniform hard capacities. *Proceedings of the 42nd IEEE Symposium on the Foundations of Computer Science*, 2001.
[26] D. Shmoys and E. Tardos. An approximation algorithm for the generalized assignment problem. *Proceedings of the 4th Annual ACM-SIAM SODA*, 1993.
[27] D. B. Shmoys, É. Tardos, and K. Aardal. Approximation algorithms for facility location problems. *Proceedings of 29th ACM STOC*, 1997.
[28] A. Srinivasan and C. Teo. A constant-factor approximation algorithm for packet routing and balancing vs. global critiria. *SIAM Journal of Computing, 30(6)*, 2001.
[29] M. Sviridenko. An 1.67-approximation algorithm for the metric uncapacitated facility lo cation problem. *Unpublished Manuscript*, 2001.
[30] L. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 1979.

APPENDIX

A. HOEFFDING-CHERNOFF BOUND

We will prove the *Hoeffding-Chernoff bound* from the *Analysis of the randomized rounding* section. First let's state a theorem due to Hoeffding

THEOREM A.1 (HOEFFDING). *Let $X_1, X_2, \dots, X_n$ be independent random variables such that $X_i \in [0,1]$ for $1 \le i \le n$. Also let $S = \sum_i X_i$, $\mu = \sum_i \mathbf{E}[X_i]$ and $0 < t < n - \mu$ then*

$$\Pr(S - \mu \ge t) \le \left(\frac{\mu}{\mu + t}\right)^{\mu+t} \left(\frac{n-\mu}{n-\mu-t}\right)^{n-\mu-t}.$$

We will use the above theorem by setting $t = \varepsilon\mu$ where $0 < \varepsilon < 1$. Thus the left hand side becomes

$$\left(\frac{1}{1+\varepsilon}\right)^{(1+\varepsilon)\mu} \left(\frac{n-\mu}{n-\mu(1+\varepsilon)}\right)^{n-\mu(1+\varepsilon)}.$$

Now we have

$$\left(\tfrac{1}{1+\varepsilon}\right)^{(1+\varepsilon)\mu} = exp(-(1+\varepsilon)\cdot\ln(1+\varepsilon)\cdot\mu) = \\ = exp\left(-\mu\cdot(\varepsilon + \tfrac{\varepsilon^2}{2} - \tfrac{\varepsilon^3}{6} + \tfrac{\varepsilon^4}{12} - \dots)\right).$$

It is easy to see that

$$\varepsilon^2 \cdot \sum_{n=1}^{\infty} \frac{(-1)^{n-1}\varepsilon^n}{(n+1)(n+2)} < \frac{\varepsilon^2}{6}$$

Thus we have

$$\left(\frac{1}{1+\varepsilon}\right)^{(1+\varepsilon)\mu} < e^{(-\mu\cdot(\varepsilon+\frac{\varepsilon^2}{3}))}.$$

On the other hand

$$\left(\tfrac{n-\mu}{n-\mu(1+\varepsilon)}\right)^{n-\mu(1+\varepsilon)} = \\ \left(1 + \tfrac{\varepsilon\mu}{n-\mu(1+\varepsilon)}\right)^{n-\mu(1+\varepsilon)} \le e^{\varepsilon\mu}.$$

Combining the results above we get

$$\Pr(S - \mu \ge \varepsilon\mu) \le e^{-\frac{\mu\varepsilon^2}{3}}.$$

Same type of calculation gives the other tail inequality

$$\Pr(S \le (1-\varepsilon)\mu) \le e^{-\frac{\mu\varepsilon^2}{2}}.$$

Which concludes the proof of the *Hoeffding-Chernoff bound* that we used.