Distributing Content Simplifies ISP Traffic Engineering

Abhigyan Sharma†, Arun Venkataramani†, Ramesh K Sitaraman†∗
† University of Massachusetts Amherst • Akamai Technologies
140 Governors Drive
Amherst MA 01003
{abhigyan,arunramesh}@cs.umass.edu

1. INTRODUCTION

Content delivery networks (CDNs) today provide a core service that enterprises use to deliver web content, downloads, and streaming media to a global audience of end-users. The traditional and somewhat simplified, tripartite view of content delivery involves three sets of entities as shown in Figure 1. The content providers (e.g., media companies, news channels, e-commerce sites, software distributors, enterprise portals, etc.) own the content and wish to provide a high-quality experience to end-users who access their content. The networks (e.g., telcos such as AT&T, multi-system operators such as Comcast, and ISPs) own the underlying network infrastructure and are responsible for provisioning capacity and routing traffic demand. Finally, the CDNs (e.g., Akamai [28], Limelight) optimize content delivery to end-users on behalf of the content providers, residing as a global, distributed overlay service [13].

Recent powerful trends are reshaping the simplified tripartite view of content delivery. A primary driver is the torrid growth of video [27, 11] and downloads traffic on the Internet. For example, a single, popular TV show with 50 million viewers, with each viewer watching an HD-quality stream of 10 Mbps, generates 500 Tbps of network traffic! The increasing migration of traditional media content to the Internet and the consequent challenges of scaling the network backbone to accommodate that traffic has necessitated the evolution of network CDNs (or NCDNs) that vertically integrate CDN functionality such as content caching and redirection with traditional network operations [19, 26, 24, 8, 36] (refer Figure 1). Another economic driver of NCDNs is the desire of networks to further monetize the “bits” that flow on their infrastructure and to offer value-added services to their own end-user subscribers, e.g., Verizon’s recent offering that delivers HBO’s content to FIOS subscribers [35]. Two key trends in how networks implement NCDNs [3] are managed CDNs where a CDN provider (such as Akamai) deploys their own servers at the network’s PoPs and operates the CDN service on behalf of the network provider [2]; and, licensed CDNs where the CDN provider licenses their software to the network who then deploy and operate the CDN themselves [1, 10].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGMETRICS’13, June 17–21, 2013, Pittsburgh, PA, USA.
Copyright 2013 ACM 978-1-4503-1900-3/13/06 ...$15.00.

†NCDNs are also called Telco CDNs, or Carrier CDNs.
As NCDNs control both the content distribution and the network infrastructure, the costs and objectives of their interest are different both from a traditional CDN and a traditional ISP. In particular, an NCDN is in a powerful position to place content in a manner that “shapes” the traffic demand so as to optimize both network cost and user-perceived latency. Indeed, several recent works have alluded to the benefits of such joint optimization strategies in the context of cooperative or competitive interaction between ISPs and content providers [38, 14, 21, 18]. NCDNs today largely treat the content distribution layer and the traffic engineering layer separately, treating the former as an overlay over the latter. However, an NCDN is the perfect setting for fielding a joint optimization as both layers are for the first time closely aligned in terms of both the business objectives and the system architecture.

The intriguing prospect of jointly optimizing content delivery and traffic engineering raises several research questions that form the focus of this paper. How should an NCDN determine content placement, network routing, and request redirection decisions so as to optimize network cost and user-perceived latency? How much benefit do joint optimization strategies yield over simpler strategies as practiced today, and does the benefit warrant the added complexity? How do content demand patterns and placement strategies impact network cost? How do planned strategies (i.e., using knowledge of recently observed demand patterns or hints about anticipated future demands) for placement and routing compare against simpler, unplanned strategies?

Our primary contribution is to empirically analyze the above questions for realistic content demand workloads and ISP topologies. To this end, we collect content request traces from Akamai, the world’s largest CDN today. We focus specifically on on-demand video and large-file downloads traffic as they are two categories that dominate overall CDN traffic and are significantly influenced by content placement strategies. Our combined traces consist of a total of 28.2 million requests from 7.79 million unique users who downloaded a total of 1455 Terabytes of content across the US over multiple days. Our main finding based on trace-driven experiments using these logs and realistic ISP topologies is that simple, unplanned strategies for placement, routing, and redirection of NCDN content are better than sophisticated joint-optimization approaches. Specifically,

- For NCDN traffic, simple unplanned schemes for placement and routing (such as least-recently-used and InverseCap) yield significantly lower (2.2–17×) network cost and user-perceived latency than a joint-optimal scheme with knowledge of the previous day’s demand.

- NCDN traffic demand can be “shaped” by simple placement strategies so that traffic engineering, i.e., optimizing routes with knowledge of recent traffic matrices, hardly improves network cost or user-perceived latency over unplanned routing (InverseCap).

- For NCDN traffic, unplanned placement and routing is just 1%-18% sub-optimal compared to a joint-optimal placement and routing with perfect knowledge of the next day’s demand at modest storage ratios (≈ 4).

2We use the term “optimal” when placement or routing is the solution of an optimization problem, but the solution may not have the lowest cost (for reasons detailed in §5.3.1)

Figure 2: NCDN Architecture

- With a mix of NCDN and transit traffic, traffic engineering does lower network cost (consistent with previous studies), but the value of traffic engineering substantially diminishes as the relative volume of NCDN traffic begins to dominate that of transit traffic.

In the rest of this paper, we first overview the NCDN architecture highlighting why it changes traditional ISP and CDN concerns (§2). Next, we formalize algorithms that jointly optimize content placement and routing in an NCDN (§3). We then describe how we collected real CDN traces (§4) and evaluate our algorithms using these traces and real ISP topologies (§5). Finally, we present related work (§7) and conclusions (§8).

2. BACKGROUND AND MOTIVATION

A typical NCDN architecture, as shown in Figure 2, resembles the architecture of a global CDN but with some important differences. First, the content servers are deployed at points-of-presence (PoPs) within a single network rather than globally across the Internet as the NCDN is primarily interested in optimizing content delivery for its own customers and end-users. Second, and more importantly, content distribution and network operations are tightly aligned, so that a joint optimization of these layers is feasible. In fact, in some cases such as a Licensed CDN, a single entity may own and manage both the content servers and the underlying network. Content providers whose content is delivered by the NCDN publish their content to origin servers that they maintain external to the NCDN itself.

Each PoP is associated with a distinct set of end-users who request content such as web, video, downloads etc. An end-user’s request is first routed to the content servers at the PoP to which the end-user is connected. If a content server at that PoP has the requested content in their cache, it serves that to the end-user. Otherwise, if the requested content is cached at other PoPs, the content is downloaded from a nearby PoP and served to the end-user. If the content is not cached in any PoP, it is downloaded directly from the content provider’s origin servers.

2.1 Why NCDNs Change the Game

Managing content distribution as well as the underlying network makes the costs and objectives of interest to an NCDN different from that of a traditional CDN or a traditional ISP. Figure 3 (top) shows the traditional concerns of content distribution and traffic engineering as addressed by a traditional CDN and a traditional ISP respectively, while Figure 3 (bottom) shows the combined concerns that an NCDN must address. We explain these in detail below.
Content Distribution (CDN) \\
Content Matrix | Storage Capacities | Traffic Matrix | Link Capacities \\
Content Placement | Request Redirection | Traffic Engineering | (Network) \\
Routing \\
NCDN Management (Network CDN) \\
Content Matrix | Storage Capacities | Link Capacities \\
Content Placement | Request Redirection | Routing

Figure 3: (Top) Traditional formulation with content distribution and traffic engineering optimized separately. (Bottom) Our new formulation of NCDN management as a joint optimization.

2.1.1 Content Distribution

A traditional CDN has two key decision components—content placement and request redirection—that seek to optimize the response time perceived by end-users and balance the load across its content servers. Content placement decides which objects should be cached at which nodes. An object may be stored at multiple nodes in the network or not stored in the network at all and be served from the origin server instead. Request redirection determines which server storing a replica of the object is best positioned to serve it.

Content placement schemes can either be planned or unplanned. A planned scheme calculates placement using a content matrix that specifies the demand for each content at each location. The content matrix is learned by monitoring a recent history of system-wide requests and possibly including hints, if any, from content providers about anticipated demand for some objects. A planned scheme uses a recent content matrix to decide on a placement periodically (e.g., daily) but does not alter its placement in between. In contrast, an unplanned scheme can continually alter its placement to accommodate an additional copy of the object at its cache to make room for new ones.

2.1.2 Traffic Engineering

A key component of ISP network operations is traffic engineering, which seeks to route the traffic demands through the backbone network so as to balance the load and mitigate hotspots. Traffic engineering is commonly viewed as a routing problem that takes as input a traffic matrix, i.e., the aggregate flow demand between every pair of PoPs observed over a recent history, and computes routes so as to minimize a network-wide cost objective. The cost seeks to capture the severity of load imbalance in the network and common objective functions include the maximum link utilization (MLU) or a convex function (so as to penalize higher utilization more) of the link utilization aggregated across all links in the network [16]. ISPs commonly achieve the computed routing either by using shortest-path routing (e.g., the widely deployed OSPF protocol [16]) or by explicitly establishing virtual circuits (e.g., using MPLS [15]). ISPs perform traffic engineering at most a few times each day, e.g., morning and evening each day [17].

Routing can also be classified as planned or unplanned in a spirit similar to content placement. Traffic engineering schemes as explained above are implicitly planned as they optimize routing for recently observed demand. To keep the terminology simple, we also classify online traffic engineering schemes [22, 15] (that are rarely deployed today) as planned. In contrast, unplanned routing schemes are simpler and rely upon statically configured routes [7, 9], e.g., InverseCap is a static shortest-path routing scheme that sets link weights to the inverse of their capacities; this is a common default weight setting for OSPF in commercial routers [17].

2.1.3 NCDN Management

As NCDNs own and manage the infrastructure for content distribution as well as the underlying network, they are in a powerful position to control all three of placement, routing, and redirection (Figure 3). In particular, an NCDN can place content in a manner that “shapes” the traffic demands so as to jointly optimize both user-perceived latency as well as network cost.

To appreciate how placement can shape traffic, consider the simple example in Figure 4. Node C has an object in its cache that is requested by end-users at nodes A and D. Suppose that one unit of traffic needs to be routed from C to A and 0.5 units from C to D to satisfy the demand for that object. The routing that achieves the minimum MLU of 0.5 to serve the demanded object is shown in the figure. Note that the routing that achieves the MLU of 0.5 is not possible with a simple, unplanned protocol like InverseCap as that would route all the traffic demand from C to A via B, resulting in an MLU of 1. Thus, a (planned) traffic engineering scheme is necessary to achieve an MLU of 0.5.

On the other hand, NCDNs can shape the traffic demand matrix by using a judicious placement and redirection strategy. Suppose that there is some space left in the content server’s cache at node B to accommodate an additional copy of the demanded object. By creating an additional copy of the object at B, the traffic demand of A can be satisfied from B and the demand of D from C achieving an MLU of 0.125. In this case, judicious content placement decreased the MLU by a factor of 4. Even more interestingly, this best MLU can be achieved using a simple routing scheme like InverseCap while also improving user-perceived latency (assuming that the latency of link BA is lower than that of the two-hop paths from C to A).

The above toy example suggests benefits to jointly optimizing placement, routing, and redirection, but raises several natural questions. How much additional benefit does such joint optimization offer compared to treating CDN and ISP concerns independently as practiced today? Is the added complexity of joint optimization strategies worth the benefit? Which of the three—placement, routing, and redirection—is the most critical to reducing network cost and user-perceived latency? How sensitive are these findings to characteristics of the content workload (e.g., video vs. download traffic)?

3. NCDN MANAGEMENT STRATEGIES

To answer the above questions, we develop an optimization model for NCDNs to decide placement, routing, and
are not concerned with traffic engineering links outside the NCDN, we model the edges \((x, a)\), for all \(x \in X\), as having infinite capacity. The virtual origin node \(o\) always maintains a copy of all the requested content. However, a request for a content is served from the virtual origin node only if no copy of the content is stored at any node \(i \in V\). In this case, the request is assumed to be routed to the virtual origin via the exit node closest to the node where the request was made (in keeping with the commonly practiced early-exit or hot potato routing policy).

ISP networks carry transit traffic in addition to NCDN traffic, which can be represented as a transit traffic matrix (TTM). Each entry in the TTM contains the volume of transit traffic between two PoPs in the network.

### 3.2 Cost Functions

We evaluate NCDN-management strategies based on two cost functions. The first cost function is maximum link utilization (or MLU) which measures the effectiveness of traffic engineering in an NCDN. MLU is a widely used network cost function for traditional TE.

The second cost function models user-perceived latency and is defined as \(\sum_{e \in E} X_e\), where \(X_e\) is the product of traffic on link \(e\) and its link latency \(L(e)\). The latency of a link \(L(e)\) is the sum of a fixed propagation delay and a variable utilization dependent delay. For a unit flow, link latency is defined as \(L_v(u_e) = p_e(1 + f(u_e))\), where \(p_e\) is the propagation delay of edge \(e\), \(u_e\) is its link utilization, and \(f(u)\) is a piecewise-linear convex function. This cost function is similar to that used by Fortz and Thorup [16]. At small link utilizations \((< 0.6)\), link latency is determined largely by propagation delay hence \(f\) is zero. At higher link utilizations \((0.9\) and above\) an increase in queuing delay and delay caused by retransmissions significantly increase the effective link latency. The utilization-dependent delay is modeled as proportional to propagation delay as the impact of (TCP-like) retransmissions is more on paths with longer links. Since \(L_v\) is convex, a set of linear constraints can be written to constrain the value of \(X_e\) (as in [16]).

### 3.3 Optimal Strategy as MIP

We present here a joint optimization strategy for NCDN-management formulated as a MIP. This formulation takes as input a content matrix, i.e., the demand for each content at each network point-of-presence (PoP), and computes content placement, request redirection and routing that minimizes an NCDN cost function while respecting link capacity and storage constraints. The decision variables for this problem are listed in Table 1. The MIP to minimize an NCDN cost function \(C\) (either MLU or latency) is as follows:

\[
\min C
\]

subject to

\[
\sum_{j \in V} t_{ijk} + t_{iok} = T_{ik}, \quad \forall k \in K, i \in V
\]

\[
\sum_{k \in K} t_{ijk} = f_{ij}, \quad \forall j \in V - X, i \in V
\]

\[
\sum_{k \in K} t_{ijk} + \sum_{k \in K} \delta_{ij} t_{iok} = f_{ij}, \quad \forall j \in X, i \in V
\]

where \(\delta_{ij}\) is 1 if \(j\) is the closest exit node to \(i\) and 0 otherwise.

---

Table 1: List of input and decision variables for the NCDN problem formulation.

Redirection so as to optimize network cost or user-perceived latency. We formulate the optimization problem as a mixed-integer program (MIP), present hardness and inapproximability results, and discuss approximation heuristics to solve MIPs for realistic problem sizes.

### 3.1 NCDN Model

Table 1 lists all the model parameters. An NCDN consists of a set of nodes \(V\) where each node represents a PoP in the network. The nodes are connected by a set of directed edges \(E\) that represent the backbone links in the network. The set of content requested by end-users is represented by the set \(K\) and the sizes of content are denoted by \(S_k\), \(k \in K\). The primary resource constraints are the link capacities \(C_e\), \(e \in E\), and the storage at the nodes \(D_i\), \(i \in V\). We implicitly assume that the content servers at the PoPs have adequate compute resources to serve locally stored content.

A content matrix (CM) specifies the demand for each content at each node. An entry in this matrix, \(T_{ik}, i \in V, k \in K\), denotes the demand (in bytes/second) for content \(k\) at node \(i\). CM is assumed to be measured by the NCDN a priori over a coarse-grained interval, e.g., the previous day. The infrastructure required for this measurement is comparable to what ISPs have in place to monitor traffic matrices today.

Origin servers, owned and maintained by the NCDN’s content providers, initially store all content published by content providers. We model origin servers using a single virtual origin node \(o\) external to the NCDN that can be reached via a set of exit nodes \(X \subset V\) in the NCDN (Figure 2).

![Figure 4: A simple NCDN example](image-url)
Note that $\delta_{ij}$ is not a variable but a constant that is determined by the topology of the network, and hence constraint (4) is linear.

$$\sum_{p \in P(l)} f_{ijp} - \sum_{q \in Q(l)} f_{ijq} = \begin{cases} f_{ij} & \text{if } l = i, \\ -f_{ij} & \text{if } l = j, \\ 0 & \text{otherwise,} \end{cases} \quad \forall i, j, l \in V$$

where $P(l)$ and $Q(l)$ respectively denote the set of outgoing and incoming links at node $l$.

$$\sum_{i \in V, j \in V} f_{ij} \leq \alpha \times C_e, \quad \forall e \in E$$

$$\sum_{k \in K} x_{ik} S_k \leq D_i, \quad \forall i \in V$$

$$x_{ok} = 1, \quad \forall k \in K$$

$$\sum_{i \in V} x_{ik} \geq z_k, \quad \forall k \in K$$

$$x_{ik} \leq z_k, \quad \forall k \in K, i \in V$$

$$t_{ijk} \leq x_{jk} T_{ik}, \quad \forall k \in K, i \in V, j \in V \cup \{o\}$$

$$t_{ok} \leq T_{ok}(1 - z_k), \quad \forall k \in K$$

$$x_{jk}, z_k \in \{0, 1\}, \quad \forall j \in V, k \in K$$

$$f_{ijc}, t_{ijk}, t_{ok} \geq 0, \quad \forall i, j \in V, c \in E, k \in K$$

The constraints have the following rationale. Constraint (2) specifies that the total traffic demand at each node for each content must be satisfied. Constraints (3) and (4) specify that the total traffic from source $j$ to sink $i$ is the sum over all content $k$ of the traffic from $j$ to $i$ for $k$. Constraint (5) specifies that the volume of a flow coming in must equal the sum of the total traffic demand at each node for each content. Constraint (6) specifies that the total flow on a link is at most $\alpha$ times capacity. Constraint (7) specifies that the total size of all content stored at a node must be less than its disk capacity. Constraint (8) specifies that all content is placed at the virtual origin node $o$. Constraints (9) and (10) specify that at least one copy of content $k$ is placed within the network if $z_k = 1$, otherwise $z_k = 0$ and no copies of $k$ are placed at any node. Constraint (11) specifies that the flow from a source to a sink for some content should be zero if the content is not placed at the source (i.e., when $x_{jk} = 0$), and the flow should be at most the demand if the content is placed at the source (i.e., when $x_{jk} = 1$). Constraint (12) specifies that if some content is placed within the network, the traffic from the origin for that content must be zero.

Updating the content placement itself generates traffic and impacts the link utilization in the network. For ease of exposition, we have deferred a formal description of the corresponding constraints to our tech report [32]. Finally, a simple extension to this MIP presented in the tech report [32] jointly optimizes routing given a TTM as well as a CM. We have presented a CM-only formulation here as our findings (in §5) show that a joint optimization of the CM and TTM is not useful for NCDNs.

### 3.5 Approximation Techniques for MIP

As solving the MIP for very large problem scenarios is computationally infeasible, we use two approximation techniques to tackle such scenarios.

The first is a two-step local search technique. In the first step, we “relax” the MIP by allowing the integral variables $x_{jk}$ and $z_k$ to take fractional values between 0 and 1. This converts an MIP into an LP that is more easily solvable. Note also that the optimal solution of the relaxed LP is a lower bound on the optimal solution of the MIP. However, the LP solution may contain fractional placement of some of the content with the corresponding $x_{jk}$ variables set to fractional values between 0 and 1. However, in our experiments only about 20% of the variables in the optimal LP solution were set to fractional values between 0 or 1, and the rest took integer values of 0 or 1. In the second step, we search for a valid solution for the MIP in the local vicinity of the LP solution by substituting the values for variables that were set to 0 or 1 in the LP solution, and re-solving the MIP for the remaining variables. Since the number of integer variables in the second MIP is much smaller, it can be solved more efficiently than the original MIP.

The second approximation technique reduces the number of unique content in the optimization problem using two strategies. First, we discard the tail of unpopular content prior to optimization. The discarded portion accounts for only 1% of all requests, but reduces the number of content by 50% or more in our traces. Second, we sample 25% of the content from the trace and, in our experiments, select trace entries corresponding only to the sampled content. These approximations reduce the number of content from tens of thousands to under 5000. An MIP of this size can be solved using local search in an hour by a standard LP solver [20] for the ISP topologies in our experiments. To check for any unintended bias introduced by the sampling, we also performed a small number of experiments with the complete trace and verified that our findings remain qualitatively unchanged.

### 4. AKAMAI CDN TRACES

To conduct a realistic simulation of end-users accessing content on an NCDN, we collected extensive traces of video and download traffic from Akamai as described below.

**Video traces.** Videos are the primary source of traffic on a CDN and are growing at a rapid rate [27, 11]. Our video trace consists of actual end-users accessing on-demand videos on the Akamai network over multiple days. To make the traces as representative as possible, we chose content providers with a wide range of business models, including major television networks, news outlets, and movie portals. The videos in our traces include a range of video types from short-duration video (less than 10 mins) such as news clips to longer duration (30 min to 120 min) entertainment videos representing TV shows and movies. In all, our anonymized traces represent a nontrivial fraction of the overall traffic on Akamai’s media network and accounted for a total of 27 million playbacks of over 85000 videos, 738 TBytes of traffic, served to 6.59 million unique end-users around the
US. Since we only had US-based network topologies with accurate link capacity information, we restricted ourselves to US-based traffic.

We collect two sets of anonymized video traces called news trace and entertainment trace respectively. The news trace was collected from a leading news outlet for an 11-day period in Sept 2011, and consists mostly of news video clips, but also includes a small fraction of news TV shows. The entertainment trace was collected for a 6 day period in January 2012, and includes a variety of videos including TV shows, clips of TV shows, movies and movie trailers from three major content providers.

The trace collection mechanism utilized a plugin embedded in the media player that is capable of reporting (anonymized) video playback information. Our traces include a single log entry for each playback and provides time of access, user id, the location of the user (unique id, city, state, country, latitude, and longitude), the url of the content, the content provider, the total length of the video (in time and bytes), the number of bytes actually downloaded, the playback duration, and the average bitrate over the playback session.

Downloads traces. Downloads of large files over HTTP is also a large large contributor of traffic in a CDN. These include software and security updates, e.g., Microsoft’s Windows or Symantec’s security updates, as well as music, books, movies, etc. The large file downloads at Akamai typically use a client-side software called the download manager [28]. We collect extensive and anonymized access data reported from the download manager using Akamai’s NetSession interface [4] for a large fraction of content providers for a period of a month (December 2010). Our traces represent a nontrivial fraction of the overall US-based traffic on Akamai’s downloads network and accounted for a total of 1.2 million downloads, 717 TBytes of traffic, served to 0.62 million unique end-users around the US. Our traces provide a single log entry for each download and provide time of access, user id, location of the user (city, state, country, latitude, and longitude), the url identifier of the content, content provider, bytes downloaded, and file size.

Figure 5 shows the fraction of requests for new content published each day relative to the previous day for news, entertainment, and downloads traces. The news trace has up to 63% requests due to new content because the latest news clips generated each day are the most popular videos on the website. The entertainment trace also has up to 31% requests each day due to new content such as new episodes of TV shows, and the previews of upcoming TV shows. The downloads trace has only 2-3% requests due to new content on a typical day. However, on the 9th day of the trace major software updates were released, which were downloaded on the same day by a large number of users. Hence, nearly 20% requests on that day were for new content. The fraction of requests for new content impacts the performance of planned placement strategies as we show §5.

5. EXPERIMENTAL EVALUATION

We conduct trace-driven experiments to compare different NCDN-management strategies. Our high-level goal is to identify a simple strategy that performs well for a variety of workloads. In addition, we seek to assess the relative value of optimizing content placement versus routing; the value of being planned versus being unplanned and the value of future knowledge about demand.

5.1 Trace-driven Experimental Methodology

To realistically simulate end-users accessing content on an NCDN, we combine the CDN traces (in §4) with ISP topologies as follows. We map each content request entry in the Akamai trace to the geographically closest PoP in the ISP topology in the experiment (irrespective of the real ISP that originated the request). Each PoP has a content server as shown in Figure 2, and the request is served locally, redirected to the nearest (by hop-count) PoP with a copy, or to the origin as needed.

ISP topologies. We experimented with network topology maps from two US-based ISPs. First is the actual ISP topology obtained from a large tier-1 ISP in the US (referred to as US-ISP). Second is the Abilene ISP’s topology [33].

MLU computation. We compute the traffic that flow through each link periodically. To serve a requested piece of content from a PoP s to t, we update the traffic induced along all edges on the path(s) from s to t as determined by the routing protocol using the bytes-downloaded information in the trace. To compute the MLU, we partition simulation time into 5-minute intervals and compute the average utilization of each link in each 5-minute interval. We discard the values of the first day of the trace in order to warm up the caches, as we are interested in steady-state behavior. We then compute our primary metric, which is the 99-percentile MLU, as the 99th percentile of the link utilization over all links and all 5-minute time periods. We use 99-percentile instead of the maximum as the former is good proxy for the latter but with less experimental noise. Finally, for ease of visualization, we scale the 99-percentile MLU values in all graphs so that the maximum 99-percentile MLU across all schemes in each graph is equal to 1. We call this scaled MLU the normalized MLU. Note that only the relative ratios of the MLUs for the different schemes matter and scaling up the MLU uniformly across all schemes is equivalent to uniformly scaling down the network resources or uniformly scaling up the traffic in the CDN traces.

Latency cost computation. Our latency cost metric,
which models user-perceived latencies, is a sum of the latency on ISP backbone links and the latency from user to its nearest PoP. Traffic served from origin incurs an additional latency from origin to the exit locations in the network. We assume origin servers to be located close to exit locations so that latency from exit locations to origin servers is a small fraction of the overall end user latency. The latency cost of a link $e$ for a interval of a second when traffic (in bits/sec) on link $e$ is $V_e$ and link utilization is $u_e$, is calculated as $V_e \times L_e(u_e)$, where $L_e$ is the latency function defined in §3. The aggregate latency cost of a link is calculated by summing the latency costs for all 1 sec intervals during the experiment (excluding the first day). The user-to-nearest PoP latency cost is calculated by summing the traffic (in bits) requested by a user times the propagation delay to its nearest PoP for all users.

Storage. We assume that storage is provisioned uniformly across PoPs except in §5.6 where we analyze heterogeneous storage distributions. We repeat each simulation with different levels of provisioned storage. Since the appropriate amount of storage depends on the size of the working set of the content being served, we use as a metric of storage the storage ratio, or the ratio of total storage at all PoPs in the network to the average storage footprint of all content accessed in a day for the trace. The total storage across all nodes for a storage ratio of 1 is 228 GB, 250 GB, and 895 GB for news, entertainment and downloads respectively.

5.2 Schemes Evaluated

Each evaluated scheme has a content placement component and a routing component. InvCap-LRU uses LRU as the cache replacement strategy and InverseCap (with ECMP) as the routing strategy. InverseCap is a static, shortest-path routing scheme where link weights are set to the inverse of the link capacity. This scheme requires no information of either the content demand or the traffic matrix. If content is available at multiple PoPs, we choose the PoP with least hop count distance while breaking ties randomly among PoPs with same hop count distance.

We added a straightforward optimization to LRU where if a user terminates the request before 10% of the video (file) is viewed (downloaded), the content is not cached (and the rest of the file is not fetched); otherwise the entire file is downloaded and cached. This optimization is used since we observe in our traces that a user watching a video very often stops watching it after watching the initial period. A similar phenomenon is observed for large file downloads, but less frequently than video.

OptRP-LRU uses an unplanned placement, LRU, but it uses an planned, optimized routing that is updated every three hours. The routing is computed by solving a multi-commodity flow problem identical to the traditional traffic engineering problem [16]. We assume that the NCDN measures the traffic matrix over the preceding three hours and computes routes that optimize the MLU for that matrix. The matrix incorporates the effect of the unplanned placement and the implicit assumption is that the content demand and unplanned placement result in a traffic matrix that does not change dramatically from one monitoring interval to the next—an assumption that also underlies traffic engineering as practiced by ISPs today.

OptRP computes a joint optimization of placement and routing once a day based on the previous day’s content matrix using the MIP formulation of §3.3. OptRP-Future has oracular knowledge of the content matrix for the next day and uses it to calculate a joint optimization of placement, redirection and routing. OptRP and OptRP-Future are identical in all respects except that the former uses the content matrix of the past day while the latter has perfect future knowledge. These two schemes help us understand the value of future knowledge. In practice, it may be possible for an NCDN to obtain partial future knowledge placing it somewhere between the two extremes. For instance, an NCDN is likely to be informed beforehand of a major software release the next day (e.g., new version of the Windows) but may not be able to anticipate a viral video that suddenly gets “hot”.

To determine the value of optimizing routing alone, we study the InvCap-OptP-Future scheme. This is a variant of OptRP-Future where InverseCap routing is used and content placement is optimized, rather than jointly optimizing both. This scheme is computed using the MIP formulation in §3.3 but with an additional constraint modification that ensures that InvCap routing is implemented.

We add a suffix -L to the names of a scheme if it is optimizing for latency cost instead of MLU, e.g., OptRP-L. For all schemes that generate a new placement each day, we implement the new placement during the low-traffic period from 4 AM to 7 AM EST. This ensures that the traffic generated due to changing the content placement occurs when the links are underutilized. For these schemes, the routing is updated each day at 7 AM EST once the placement update is finished.

5.3 Comparison of Network Cost

5.3.1 Analysis of Video & Downloads Traffic

Figure 6 shows the results for the news, entertainment and downloads traces on Abilene and US-ISP. Our first observation is that a realistic planned placement and routing scheme, OptRP, performs significantly worse than a completely unplanned scheme, InvCap-LRU. OptRP has 2.2× to
Plan.

**Figure 6:** Planned OptRP performs much worse than unplanned InvCap-LRU. OptRP-Future performs moderately better than InvCap-LRU primarily at small storage ratios.

17× higher MLU than InvCap-LRU even at the maximum storage ratio in each graph. OptRP has a high MLU because it optimizes routing and placement based on the previous day’s content demand while a significant fraction of requests are for new content not accessed the previous day (see Figure 5). Due to new content, the incoming traffic from origin servers is significant, so the utilization of links near the exit nodes connecting to the origin servers is extremely high.

The fraction of requests served from the origin is much higher for OptRP compared to InvCap-LRU and OptRP-Future on the news and the entertainment traces. Figure 7 shows that OptRP serves 50% and 21% of requests from the origin for news and entertainment respectively. In comparison, InvCap-LRU and OptRP-Future serve less than 2% of requests from the origin. Therefore, OptRP has a much higher MLU than both InvCap-LRU and OptRP-Future on the two traces.

The downloads trace differs from other traces in that, except for one day, the traffic is quite predictable based on the previous day’s history. This is reflected in the performance of OptRP that performs nearly the same as OptRP-Future on all days except the ninth day of the trace (see Figure 8). The surge in MLU for OptRP on the ninth day is because nearly 20% of requests on this day is for new content consisting of highly popular software update releases (see Figure 5). The surge in MLU on this one day is mainly responsible for the poor performance of OptRP on the downloads trace.

Next, we observe that InvCap-LRU does underperform compared to OptRP-Future that has knowledge of future content demand. However, InvCap-LRU improves with respect to OptRP-Future as the storage ratio increases. The maximum difference between the two schemes is for the experiment
with entertainment trace on US-ISP topology. In this case, at a storage ratio of 1, InvCap-LRU has twice the MLU of the OptRP-Future scheme; the difference reduces to 1.6× at a storage ratio of 4. This shows that when storage is scarce, planned placement with future knowledge can significantly help by using knowledge of the global demand to maximize the utility of the storage. However, if storage is plentiful, the relative advantage of OptRP-Future is smaller. An important implication of our results is that an NCDN should attempt to do planned placement only if the future demand can be accurately known or estimated. Otherwise, a simpler unplanned scheme such as LRU suffices.

How are the above conclusions impacted if InvCap-LRU were to optimize routing or OptRP-Future were to use InverseCap routing? To answer this question, we analyze the maximum reduction in MLU by using OptRP-LRU over InvCap-LRU across all storage ratios in Figure 9. We similarly compare OptRP-Future and InvCap-OptP-Future. We find that OptRP-LRU improves the MLU over InvCap-LRU by at most 10% across all traces suggesting that optimizing routing is of little value for an unplanned placement scheme. OptRP-Future reduces the network cost by at most 13% compared to InvCap-OptP-Future. As we consider OptRP-Future to be the “ideal” scheme with full future knowledge, these results show that the best MLU can be achieved by optimizing content placement alone; optimizing routing adds little additional value.

Why do InvCap-LRU and OptRP-LRU have nearly the same network costs? While LRU does greatly reduce traffic due to a high percentage of cache hits, but this is not enough to explain why InvCap achieves nearly the same MLU as optimized routing for the residual traffic, however small. Traffic engineering gives little additional value either because traffic matrices are unpredictable and/or because the NCDN traffic matrices and ISP topologies that we consider do not give much scope for an optimized routing to reduce the MLU over InverseCap routing.

Somewhat counterintuitively, the MLU sometimes increases with a higher storage ratio for the OptRP scheme. There are three reasons that explain this. First, the optimization formulation optimizes for the content matrix assuming that the demand is uniformly spread across the entire day, however the requests may actually arrive in a bursty manner. So it may be sub-optimal compared to a scheme that is explicitly optimized for a known sequence of requests. Second, the optimization formulation optimizes the MLU for the “smoothed” matrix, but the set of objects placed by the optimal strategy with more storage may not necessarily be a superset of the objects placed by the strategy with lesser storage at any given PoP. Third, and most importantly, the actual content matrix for the next day may differ significantly from that of the previous day. All of these reasons make the so-called “optimal” OptRP strategy suboptimal and in combination are responsible for the nonmonotonicity observed in the experiments.

5.3.2 Content Chunking

Content chunking is widely used today to improve content delivery and common protocols such as HTTP [30] and Apple HLS [5] support content chunking. This experiment analyzes the effect of content chunking on our findings. In these experiments, we split videos into chunks of 5 minute duration. The size of a video chunk depends on the video bitrate. For the downloads trace, we split content into chunks of size 50 MB.

Our results show that although chunking improves performance of both InvCap-LRU and OptRP-Future, it significantly improves the performance of InvCap-LRU relative to OptRP-Future (see Figure 10). Due to chunking, the maximum difference between the MLU of InvCap-LRU and OptRP-Future reduces from 2.5× to 1.4×. At the maximum storage ratio, InvCap-LRU is at most 18% worse compared to OptRP-Future. Our experiments on other traces and topologies (omitted for brevity) show that InvCap-LRU has at most 4% higher network cost than OptRP-Future at the maximum storage ratio. An exception is the news trace, where chunking makes a small difference as more than 95% content is of duration less than our chunk size. Hence, chunking strengthens our conclusion that InvCap-LRU achieves close to the best possible network cost for an NCDN. Even with chunking, OptRP has up to 7× higher MLU compared to InvCap-LRU (not shown in Figure 10). This is because chunking does not help OptRP’s primary problem of not being able to adapt effectively to new content, so it continues to incur a high cost.

5.3.3 Alternative Planned Schemes

The experiments so far suggest that a planned scheme that engineers placement and routing once a day based on the previous day’s demand performs poorly compared to an unplanned scheme, InvCap-LRU. Hence, in this section, we evaluate the performance of two alternative planned schemes.

First, we evaluate a hybrid placement scheme, which splits the storage at each node into two parts - one for a planned placement based on the previous day’s content demand (80% of storage) and the other for placing the content in a unplanned LRU manner (20% of storage). We find that InvCap-
LRU performs either as well or better than the hybrid scheme. Assigning a greater fraction of storage to unplanned placement does not change the above conclusions (graph omitted for brevity). Of course, a carefully designed hybrid scheme by definition should perform at least as well as the unplanned and planned schemes, both of which are extreme cases of a hybrid strategy. However, we were unable to design simple hybrid strategies that consistently outperformed fully unplanned placement and routing.

Next, we analyze the performance of planned schemes that engineer placement and routing multiple times each day at equal intervals - twice/day, 4 times/day, and 8 times/day. In all cases, we engineer using the content demand in the past 24 hours. As Figure 12 shows, OptRP needs to engineer 8 times/day to match the performance of the InvCap-LRU scheme. In all other cases, InvCap-LRU performs better. In fact, the experiment shown here represents the best case for OptRP. Typically, OptRP performs worse even when engineering is done 8 times/day, e.g., on the news trace, we find OptRP incurs up to 4.5× higher MLU compared to InvCap-LRU even on engineering 8 times/day.

Executing a planned placement requires considerable effort—measuring content matrix, solving a computationally intensive optimization, and moving content to new locations. Further, a planned placement needs to be executed 8 times a day (or possibly more) even to match the cost achieved by an unplanned strategy. Our position is that NCDNs are better served by opting for a much simpler unplanned strategy and provisioning more storage, in which case, an unplanned strategy already obtains a network cost close to the best a planned strategy can possibly achieve.

### 5.4 Comparison of Latency Cost

We compare InvCap-LRU scheme, which is a completely unplanned scheme, against OptRP-L and OptRP-Future-L, which optimize latency cost based on previous day’s content matrix and based on next day’s content matrix respectively.

We experiment with ISP topologies in which links are scaled down uniformly. We needed to scale down the links as our traces did not generate enough traffic to fill even 5% of the capacity of the links during the experiment; ISP networks are unlikely to operate at such small link utilizations. The network topology is scaled such that the 99th-percentile MLU for results is 75% link utilization for the InvCap-LRU scheme. This ensures that network has sufficient capacity to support content demand at all storage ratios and network links are not heavily under-utilized.

We present the results of our comparison on the US-ISP topology in Figure 13. Experiments on the Abilene topology show qualitatively similar conclusions (graph omitted for brevity). We find that on the news and entertainment traces, OptRP-L scheme results in an order of magnitude higher latency costs. OptRP-L scheme is similar to OptRP scheme except it optimizes latency instead of network cost. Like the OptRP scheme, OptRP-L is unable to predict the popularity of new content resulting in high volume of traffic from origin servers and high link utilization values. OptRP-L either exceeds link capacities or operates close to link capacity for some links which results in very high latencies.

The latency cost of InvCap-LRU relative to OptRP-Future-L improves with an increase in storage ratio. At the smallest storage ratio, InvCap-LRU has 70-110% higher latency cost than OptRP-Future-L. The difference reduces to 14-34% at the maximum storage ratio. Higher storage ratio translate to higher cache hit rates, which reduces propagation delay of transfers and lowers link utilizations. Both these factors contribute to a smaller latency cost for InvCap-LRU. This finding shows that NCDNs can achieve close to best latency costs with an unplanned scheme InvCap-LRU and provisioning moderate amounts of storage.

The performance of OptRP-L on the downloads trace is much closer to OptRP-Future-L than on the other two traces. Unlike other traces, content popularity is highly predictable on the downloads trace based on yesterday’s demand, except for a day on which multiple new software releases were done. On all days except one, OptRP-L has nearly optimal latency cost and it incurs a higher latency cost on one day of the trace. As a result, OptRP-L’s aggregate latency cost summed over all days is only moderately higher than that of OptRP-Future-L.

### 5.5 Effect of NCDN Traffic on Network Cost

This experiment, unlike previous experiments, considers a network consisting of both ISP and NCDN traffic. Our goal is to evaluate how network costs change as the fraction of NCDN traffic increases in the network. Second, we seek to examine the benefit of optimizing routing over an unplanned routing scheme, InverseCap. To this end, we compare the performance of InvCap-LRU and OptR-LRU schemes. The latter scheme optimizes routing for the combined traffic matrix due to NCDN traffic and ISP transit traffic. In order to estimate the best gains achievable with an optimized routing, we provide to the OptR-LRU scheme knowledge of future ISP traffic matrices. OptR-LRU cannot be provided the knowledge of future NCDN traffic matrices because NCDN traffic matrices can only be measured from experiment itself and we do not know them beforehand. We optimize rout-
matrices were collected in February, 2005. Since ISP traf-
ing once a day in this experiment. Varying the frequency of 
routing update did not improve OptR-LRU’s performance.

We experiment with hourly transit traffic matrices span-
ning 7 days from the same Tier-1 ISP — US-ISP. These 
matrices were collected in February, 2005. Since ISP traf-

Figure 14: [News, US-ISP] Network costs at varying 
fractions of NCDN traffic in an ISP network.

We found that this was because the NCDN traffic showed 
highly variable link utilization even over the course of a few 
minutes: the maximum link utilization differed by up to 3× in 
the course of 15 minutes. The hourly ISP traffic matrix 
that we experimented with retained the same, smoothed uti-

To make the variability of NCDN traffic comparable to 
ISP traffic, we scaled up the volume of NCDN traffic. The 
scaling is done by introducing new content similar to a ran-
domly chosen content in the original trace. Each new con-
tent is of the same size, and same video bit rate as the origi-

request redirection strategy for InvCap-LRU that periodically 
measures link utilizations in the network and prefers less 
loaded paths while redirecting requests. Our evaluation 
shows that such a redirection gives small benefits in terms of 
network cost (7% – 13%) and gives almost no benefits on la-
tency costs. This implies that sophisticated network-aware 
redirection strategies may be of little value for an NCDN.

Request redirection to neighbors: If each PoP redi-
 rects requests only to its one-hop neighbor PoPs before redi-
recting to the origin, InvCap-LRU incurs only a moderate 
(6%-27%) increase in the MLU. However, if a PoP redirects 
to no other PoPs but redirects only to the origin, the MLU 
for InvCap-LRU increases significantly (25%-100%). Thus, 
request redirection to other PoPs helps reduce network cost, 
but most of this reduction can be had by redirecting only to 
neighboring PoPs.

Heterogenous storage: Heterogenous storage at PoPs 
(storage proportional to the number of requests at a PoP in 
a trace, and other simple heuristics) increases the MLU 
compared to homogenous storage for both InvCap-LRU and 
OptRP-Future, and makes InvCap-LRU more sub-optimal com-
pared to OptRP-Future. This leads us to conclude that our 
results above with homogenous storage are more relevant 
to practical settings.

Number of caches: If caches are deployed on all PoPs, 
MLU is significantly lower compared to scenarios when caches 
are deployed only at a fraction of PoPs: the total storage 
across PoPs is same in all scenarios. This suggests that 
NCDNs should deploy caches at all PoPs to minimize MLU.

OptR-LRU parameters: Whether OptR-LRU updates rout-
ing at faster timescales (every 15 minutes, or 30 minutes) or 
slower timescales (6 hours, or 24 hours) than the default up-
date interval of 3 hours, its performance is nearly the same. 
Further, whether OptR-LRU optimizes routing using traffic 
matrix measured over the immediately preceding three 
hours (default) or using traffic matrices measured the pre-
vious day, its network cost remains nearly unchanged. This 
reinforces our finding that optimizing routing gives minimal 

Number of exit nodes: When the number of network 
exit nodes is increased to five or decreased to one, our find-
ings in §5.3.1 remain qualitatively unchanged.

Link failures: The worst-case network cost across all 
single link failures for InvCap-LRU as well as OptRP-Future is 
approximately twice compared to their network costs dur-
ing a failure-free scenario. Comparing the failure-free sce-
nario and link failure scenarios, the relative sub-optimality of 
InvCap-LRU with respect to OptRP-Future remains the 
same at small storage ratios but reduces at higher ratios.

6. LIMITATIONS AND FUTURE WORK

Our experimental methodology suffers from some short-
comings. First, we assume that servers deployed at each 
PoP have enough resources to serve users requests for locally 
cached content. In cases when server resources are inade-

find these results to be consistent with our earlier experi-
ments with NCDN traffic only.

5.6 Other Results and Implications

We summarize our main conclusions from the rest of our 
experiments here and refer the reader to our tech report [32] 
for a complete description of these experiments:

Link-utilization aware redirection: We evaluate a re- 
quest redirection strategy for InvCap-LRU that periodically 
measures link utilizations in the network and prefers less 
loaded paths while redirecting requests. Our evaluation 
shows that such a redirection gives small benefits in terms of 
network cost (7% – 13%) and gives almost no benefits on la-
tency costs. This implies that sophisticated network-aware 
redirection strategies may be of little value for an NCDN.

Request redirection to neighbors: If each PoP redi-
 rects requests only to its one-hop neighbor PoPs before redi-
recting to the origin, InvCap-LRU incurs only a moderate 
(6%-27%) increase in the MLU. However, if a PoP redirects 
to no other PoPs but redirects only to the origin, the MLU 
for InvCap-LRU increases significantly (25%-100%). Thus, 
request redirection to other PoPs helps reduce network cost, 
but most of this reduction can be had by redirecting only to 
neighboring PoPs.

Heterogenous storage: Heterogenous storage at PoPs 
(storage proportional to the number of requests at a PoP in 
a trace, and other simple heuristics) increases the MLU 
compared to homogenous storage for both InvCap-LRU and 
OptRP-Future, and makes InvCap-LRU more sub-optimal com-
pared to OptRP-Future. This leads us to conclude that our 
results above with homogenous storage are more relevant 
to practical settings.

Number of caches: If caches are deployed on all PoPs, 
MLU is significantly lower compared to scenarios when caches 
are deployed only at a fraction of PoPs: the total storage 
across PoPs is same in all scenarios. This suggests that 
NCDNs should deploy caches at all PoPs to minimize MLU.

OptR-LRU parameters: Whether OptR-LRU updates rout-
ing at faster timescales (every 15 minutes, or 30 minutes) or 
slower timescales (6 hours, or 24 hours) than the default up-
date interval of 3 hours, its performance is nearly the same. 
Further, whether OptR-LRU optimizes routing using traffic 
matrix measured over the immediately preceding three 
hours (default) or using traffic matrices measured the pre-
vious day, its network cost remains nearly unchanged. This 
reinforces our finding that optimizing routing gives minimal 

Number of exit nodes: When the number of network 
exit nodes is increased to five or decreased to one, our find-
ings in §5.3.1 remain qualitatively unchanged.

Link failures: The worst-case network cost across all 
single link failures for InvCap-LRU as well as OptRP-Future is 
approximately twice compared to their network costs dur-
ing a failure-free scenario. Comparing the failure-free sce-
nario and link failure scenarios, the relative sub-optimality of 
InvCap-LRU with respect to OptRP-Future remains the 
same at small storage ratios but reduces at higher ratios.

6. LIMITATIONS AND FUTURE WORK

Our experimental methodology suffers from some short-
comings. First, we assume that servers deployed at each 
PoP have enough resources to serve users requests for locally 
cached content. In cases when server resources are inade-
quate, e.g., due to flash crowds, a simple redirection strategy, e.g., redirection to the closest hop-count server used by InvCap-LRU, may result in poor user-perceived performance. In practice, NCDNs should adopt a redirection strategy that takes server load into account to handle variability of user demands. Second, we measure latency using a utility-based cost function that can be efficiently computed using flow-level simulations. An evaluation of end-user perceived metrics, e.g., TCP throughput, would be more convincing, but requires a measurement-based evaluation or a packet-level simulation. A measurement-based evaluation requires network and server infrastructure similar to an NCDN, which is beyond our resources. Even packet-level simulations become extremely time consuming at the scale of an ISP network, which we observed in an earlier work [31]. Third, the latency comparison is done for large, static objects and is not generalizable to dynamic content and small objects. We defer addressing these concerns to future work.

Another open question is whether our conclusions are generalizable for other topologies and workloads. For instance, our preliminary analysis with a synthetic workload trace (included in [32]) suggests that the InvCap-LRU scheme may not give the close to optimal costs in all scenarios. An evaluation of relative performance of schemes for general topologies and workloads would be considered in our future work.

7. RELATED WORK

Traffic engineering and content distribution have both seen an enormous body of work over more than a decade. To our knowledge, our work is the first to consider the NCDN problem, wherein a single entity seeks to address both concerns, and empirically evaluate different placement, routing, and redirection strategies.

**Joint optimization:** Recent work has explored the joint optimization of traffic engineering and “content distribution”, where the latter term refers to the server selection problem. P4P (Xie et al. [38]) shows that P2P applications can improve their performance and ISPs can reduce the MLU and interdomain costs, if P2P applications adapt their behavior based on hints supplied by ISPs. Jiang et al. [21] and DiPalantino et al. [14] both study the value of joint optimization of traffic engineering and content distribution versus independent optimization of each. CaTE (Frank et al. [18]), like P4P, shows that a joint optimization can help both ISPs and content providers improve their performance. Valancius et al. [34] propose a system which helps online service providers choose the best server replica for each client considering multiple server replicas and multiple network paths to each replica. Further, they quantify the benefit of this “joint routing” approach over “content routing”, i.e., choosing best replica with only a single path to each replica, and over “network routing”, i.e., choosing best path to an unreplicated server among multiple paths. Xu et al. [90] study a similar problem. These works equate content distribution to server selection (or request redirection in our parlance), while the NCDN problem additionally considers content placement itself as a degree of freedom. We find that the freedom to place content is powerful enough that even unplanned placement and routing strategies suffice to achieve close to best latency and network costs for NCDNs, making joint optimization of content distribution and traffic engineering unnecessary.

**Placement optimization:** In the context of CDNs, many variants of content or service placement problems have been studied [29, 25, 12, 23]. A recent work is that of Applegate et al. [6], who study the content placement problem for a VoD system that seeks to minimize the aggregate network bandwidth consumed. However, they assume a fixed routing in the network, while one of our contributions is to assess the relative importance of optimizing routing and optimizing placement in an NCDN.

Furthermore, they find that an optimized, planned placement with a small local cache (similar to our “hybrid” strategy in §5.3.3) outperforms LRU. In contrast, our experiments suggest otherwise. There are three explanations for this disparity. First, their workload seems to be predictable even at weekly time scales, whereas the Akamai CDN traces that we use show significant daily churn. Second, their scheme has some benefit of future knowledge and is hence somewhat comparable to our OptRP-Future. For a large NCDN, obtaining knowledge about future demand may not be practical for all types of content, e.g., breakout news videos. Finally, our analysis suggests that LRU performs sub-optimally only at small storage ratios, and the difference between LRU and OptRP-Future reduces considerably at higher storage ratios (not considered in [6]).

**Traffic engineering:** Several classes of traffic engineering schemes such as OSPF link-weight optimization [16], MPLS flow splitting [15], optimizing routing for multiple traffic matrices [37, 40], online engineering [22, 15], and oblivious routing [7, 9], have been studied. All of these schemes assume that the demand traffic is a given to which routing must adapt. However, we find that an NCDN is in a powerful position to change the demand traffic matrix, so much so that even a naive scheme like InverseCap, i.e., no engineering at all, suffices in conjunction with a judicious placement strategy and optimizing routing further adds little value. In this respect, our findings are comparable in spirit to Sharma et al. [31]. However, they focus on the impact of location diversity, and show that even a fixed number of randomly placed replicas of each content suffice to blur differences between different engineering schemes with respect to a capacity metric (incomparable to MLU), but find that engineering schemes still outperform InverseCap.

8. CONCLUSIONS

We posed and studied the NCDN-management problem where content distribution and traffic engineering decisions can be optimized jointly by a single entity. Our trace-driven experiments using extensive access logs from the world’s largest CDN and real ISP topologies resulted in the following key conclusions. First, simple unplanned schemes for routing and placement of NCDN content, such as InverseCap and LRU, outperform sophisticated, joint-optimal placement and routing schemes based on recent historic demand. Second, NCDN traffic demand can be “shaped” by effective content placement to the extent that the value of improving routing for NCDN traffic is small. Third, we studied the value of the future knowledge of demand for placement and routing decisions. While future knowledge helps, what is perhaps surprising is that a small amount of additional storage allows simple, unplanned schemes to perform as well as planned ones with future knowledge. Finally, with a mix of NCDN and transit traffic, the benefit of traditional traffic engineering is commensurate to the fraction of traffic that is transit traffic, i.e., ISPs dominated by NCDN traffic.
can simply make do with static routing schemes. Overall, our findings suggest that content placement is a powerful degree of freedom that NCDNs can leverage to simplify and enhance traditional traffic engineering.

9. ACKNOWLEDGEMENTS

We sincerely thank our shepherd Richard Ma and the anonymous reviewers for their feedback. This research was supported by following grants from National Science Foundation: CNS-0845855, CNS-0917078, and CNS-1040781. We thank S. Shunmuga Krishnan, Yin Lin, and Bruce M. Maggs for their help with the Akamai trace data collection. Any opinions expressed in this work are solely those of the authors and not necessarily those of National Science Foundation or Akamai Technologies.

10. REFERENCES

[1] Akamai Licensed CDN. http://www.akamai.com/html/solutions/auraLicensed_cdn.html.

[2] Akamai Managed CDN. http://www.akamai.com/html/solutions/auraManaged_cdn.html.

[3] Akamai. AT&T strike deal for content delivery services, December 2012. http://www.marketresearch.com/story/akamai-at-strike-deal-for-cdn-services-2012-12-06.

[4] A Elwalid, C Jin, S Low, and I Widjaja. MATE: MPLS Traffic Engineering vs. Content Engineering. In INFOCOM, 2009.

[5] B Fortz and M Thorup. Optimizing OSPF weights in a changing world. SIGCOMM, 2002.

[6] D Applegate, A Archer, V Gopalakrishnan, S Lee, and K K Ramakrishnan. Optimal content placement for a large-scale VoD system. In CoNEXT, 2010.

[7] D Applegate and E Cohen. Making routing robust to changing traffic demands: algorithms and evaluation. IEEE/ACM Trans. Netw., 14:1193–1206, December 2006.

[8] A Elwalid and G Nakibly. A traffic engineering approach for placement and selection of network services. IEEE/ACM Trans. Netw., 2009.

[9] J. Dilley, B. Maggs, J. Pariksh, H. Prokop, R. Sitaraman, and B. Weihl. Globally distributed content delivery. Internet Computing, IEEE, 6(5):50–58, 2002.

[10] D DiPalantino and R. Johari. Traffic Engineering vs. Content Distribution: A Game Theoretic Perspective. In INFOCOM, 2009.

[11] C Zhang, Y Liu, W Gong, J Kurose, R Moll, and D Towsley. On optimal routing with multiple traffic matrices. In INFOCOM, 2005.

APPENDIX

A. COMPLEXITY OF NCDN PROBLEM

Opt-NCDN is the decision version of the NCDN problem (§3). Opt-NCDN asks if the MLU of the network can be α while satisfying the constraints of the problem.

THEOREM 1 Opt-NCDN is NP-Complete even in the special case where all objects have unit size, all demands, and link and storage capacities have binary values.

Proof: We show a reduction from the well known SetCover problem defined as follows. SetCover: Let $S = \{1, 2, ..., n\}$ be a set of n elements. Let $X = \{S_1, ..., S_m\}$ where $S_i \subseteq S, 1 \leq i \leq m$. Let k be an integer. SetCover asks if there exists $Y = \{Y_1, ..., Y_k\}$, where $Y_i \subseteq X$ and $S = \bigcup Y_i$. Set $Y$ is called a set cover of size k.

The reduction from SetCover to Opt-NCDN is described using the network in Figure 15. Set $V_1 = \{1, ..., m\}$ refers to nodes in the top row. Each node i in $V_1$ maps to the set $S_i \subseteq S$. Set $V_2 = \{1, ..., n\}$ refers to nodes in the bottom row excluding node s. Each node i in $V_2$ maps to element i in $S$. Node s is called a special node.

Directed links $(i, j)$ exist from all nodes i in $V_1$ to all nodes $j \in V_2$. The capacity of $(i, j)$ is 1 unit if $i \in S_j$, otherwise capacity is zero. Node s has incoming links $(i, s)$ from all nodes i in $V_1$ such that the capacity of all incoming links is 1 unit. All nodes in the top row $V_1$ have unit storage whereas nodes in the bottom row $V_2 \cup \{s\}$ have zero storage.

The set of objects is $\{0, 1, 2, ..., (m - k)\}$ and all objects have unit size. Object o is a special object that has unit demand at nodes in set $V_2 = \{1, ..., n\}$ and zero demand at all other nodes. Objects 1, 2, ..., $(m - k)$ have unit demand at special node s and zero demand at all other nodes.

CLAIM: There is a set cover of size k if and only if the above network can achieve $\mu = 1$.

If there is a set cover of size k, then the network can achieve $\mu = 1$. Store the special object o at the k set cover locations in the top row and satisfy demand for o at
A node for each set

These nodes have unit storage but no demand for any content

A node for each element

These nodes have no storage but have non-zero demand for content

Unit capacity edges from all top nodes to all bottom nodes.

Figure 15: Reduction from SetCover to Opt-NCDN

Nodes $V_2 = \{1, ..., n\}$ in the bottom row from these locations with MLU = 1. The remaining $(m - k)$ nodes in the top can be used for objects $\{1, 2, ..., (m - k)\}$ to satisfy the demand at special node $s$ with MLU of 1.

If there is no set cover of size $k$, then the network must have a MLU $> 1$. Objects must be placed in some $(m - k)$ nodes in the node $V_1 = \{1, ..., m\}$ in the top row to satisfy the demand for special node $s$. Thus, at most $k$ nodes are available for placing special object $o$. Since there is no set cover of size $k$, some bottom node $i \in V_2$ must satisfy its demand for special object $o$ using an edge whose capacity is zero resulting in MLU = $\infty$ on that edge.

It is easy to show that Opt-NCDN $\in$ NP. Hence, Opt-NCDN is NP-Complete.

**Theorem 2** Opt-NCDN is inapproximable within a factor $\beta$ for any $\beta > 1$ unless $P = NP$.

The proof of Theorem 1 shows that if there is a set cover of size $k$, MLU = 1 and MLU = $\infty$ otherwise. Thus, if we find a solution for which MLU is finite, it implies that MLU = 1, which immediately gives a solution to the corresponding SetCover instance.

Let's assume a $\beta$-approximation ($\beta > 1$) exists for Opt-NCDN. Then, we can solve SetCover in polynomial time by mapping SetCover instance to Opt-NCDN instance, and checking if MLU $\leq \beta$ (which implies MLU = 1). As SetCover $\in$ NP-Complete, therefore, no $\beta$-approximation for Opt-NCDN exists unless $P = NP$. 