To Coordinate Or Not To Coordinate? Wide-Area Traffic Management for Data Centers

Srinivas Narayana†, Joe Wenjie Jiang†, Jennifer Rexford†, Mung Chiang*
†Department of Computer Science, and *Department of Electrical Engineering
Princeton University
{narayana, wenjiej, jrex, chiangm}@princeton.edu

ABSTRACT

In this paper, we study the impact of coordinating the selection of data centers for clients ("mapping") and performing multi-homed network-routing ("routing") from data centers, two decisions which are conventionally managed independently. We model their functional separation and degrees of coordination through an optimization framework, and study the impact of coordination on (i) service performance, (ii) robustness to traffic variability, and (iii) bandwidth costs. We show that in theory, performing mapping and routing independently can lead to much lower performance or higher costs than a coordinated decision. In practice, our trace-based evaluations on an operational CDN show that coarse-grained information-sharing between mapping and routing is sufficient for near-optimal request latencies, but not minimal costs. Further, even complete information-sharing between mapping and routing is not sufficiently robust to traffic variability, as ISP-links can easily be overwhelmed due to traffic burstiness. To address this issue, we design a coordination technique which is much more robust to traffic variability, and is also provably optimal.

1. INTRODUCTION

Organizations running online services worry about the end-to-end performance their customers experience, since even small increases in perceived latency can have significant impact on revenue [1]. To improve performance and reliability, cloud service providers (CSPs) typically run multiple geographically-distributed data centers (DCs), each peering with multiple ISPs. Placing servers closer to users reduces propagation delay, and path diversity resulting from multi-homing offers performance benefits [2]. However, service providers also need to consider operational costs—*e.g.,* large services may send and receive petabytes of data in a day, leading to significant bandwidth costs [3], and can spend tens of millions of dollars annually on electricity [4].

Client demands and path performance, and even electricity and bandwidth costs, vary over time. As such, CSPs should adapt how they direct client requests to data centers (*client mapping*) and response traffic to wide-area paths (*network routing*). A CSP with its own backbone network could direct clients to the nearest ingress point (*e.g.,* a front-end proxy [5]), and then optimize wide-area routing across all upstream ISPs [3]. However, many CSPs connect their data centers directly to the Internet, and rely on mechanisms like DNS or HTTP redirection to map client requests to a particular data center. Then, the data center's edge router selects a local upstream ISP to carry response traffic back to the client. Together, these mapping and routing decisions determine whether the CSP offers good performance and balanced loads, at a reasonable cost.

Today, request-mapping and response-routing decisions are made *independently*, which can in principle create situations leading to bad performance and high costs (§2). For example, the mapping system could easily direct too many requests to a data center with limited upstream bandwidth, poor performance [6] or expensive ISP connectivity, leaving the routing system no ability to rectify the problem. This incites the question of whether *improved mutual visibility* between the mapping and routing systems can indeed boost performance and reduce costs in practice. Hence, we ask to what extent CSPs can benefit from *coordination of currently existing* mapping and routing (before building new centralized systems), and to understand *which information is most valuable* to share in order to arrive at good collective decisions.

In this paper, we focus on understanding the implications of coordinating mapping and routing decisions for service performance, robustness to traffic variability, and cost, when the mapping system has varying levels of visibility into routing. We employ optimization models for mapping-routing schemes with different levels of visibility, and realistic CDN traffic traces for evaluations. Motivated by robustness considerations from our evaluations, we propose a mapping-routing coordination scheme, and leverage results from nonconvex optimization [7] to show that this scheme provably converges to an optimum solution. We make the following contributions:

A case for coordinated mapping and routing. In §2 we present toy examples that illustrate how in-

dependent control of mapping and routing can lead to lower performance and higher costs. We also show scenarios where separately optimizing mapping and routing decisions in an iterative fashion can be detrimental.

Optimization formulations to model coordination. In §3 we formulate a set of optimization models that capture different levels of mapping's visibility into routing—broadly, per-data center and per-path information—that progress from least to most coordinated mapping schemes. Our formulation also captures other practical considerations like path performance, link capacities, client demands, and bandwidth costs.

Coarse-grained information-sharing is sufficient for latency optimization. In §4 we show that sharing per-data center information—namely per-client smallest latency and aggregate DC bandwidth—with the mapping system is sufficient for near-optimal propagation delays. The reason is that geographic proximity dominates propagation delays—*i.e.*, latency differences between paths from different DCs greatly outweigh the differences between paths from the same DC.

Even fully coordinated mapping-routing may not be robust to traffic variability. In §5 we show that even with complete routing-system visibility, several links can have high utilizations due to traffic burstiness between successive optimization time-intervals, as well as from inherent variability [8] within an interval.

Robust, provably optimal mapping-routing with utilization penalties. Our insight (§6) is that explicitly penalizing high link utilizations helps leave sufficient bandwidth headroom to absorb traffic bursts. This approach avoids a tricky tradeoff between performance loss and high utilization if a fixed "burst capacity" reservation is used, while also achieving provable optimality.

Sharing per-link costs is important for good costs overall. In §7 we show that visibility into per-link bandwidth costs is important to minimize costs.

The rest of this paper is organized as follows. §2 reasons why coordinating mapping and routing could be useful, with toy examples. §3 formulates coordination and its various forms into optimization models. §4 evaluates the impact of coordination on propagation delay, and §5 on robustness to traffic variability. §6 presents a robust, provably optimal mapping-routing scheme. §7 evaluates the impact of visibility on overall costs. §8 discusses why our results are generalizable. §9 discusses the related work and §10 concludes.

2. COORDINATING MAPPING & ROUTING

Performing client-mapping and network-routing independently can miss opportunities to improve performance or reduce costs. Through toy examples consisting of two data centers and one set of users *e.g.,* a single IP prefix, we highlight some challenges that can hinder independent mapping and routing optimizations from arriving at good collective decisions.

Misaligned objectives can lead to suboptimal decisions. Typically, mapping and routing systems work with different objectives—*e.g.,* mapping performs latency and load-based DC selection, while routing considers end-to-end performance and bandwidth costs to choose a peer to forward responses. Due to mismatched objectives, overall performance and cost can both suffer. In Fig. 1(a), the routing system at each DC picks the peer with the least cost per unit bandwidth, while the mapping system picks the DC with the least propagation delay given current routing choices—resulting in a situation where the service neither has globally optimal cost nor optimal latency.

Incomplete visibility can lead to suboptimal decisions. In Fig. 1(b), the mapping and routing systems both optimize latency, and users are sending traffic at the rate of 5Gb/s. Without information on link loads and capacities, the mapping system directs too much traffic to the DC on the left, leading to large queueing delays and packet losses. If the mapping system uses information about link capacities, it could easily direct most traffic to the alternate DC with ample spare capacity, and only slightly worse latency.

Coupled operational constraints can lead to suboptimal equilibria. Even if mapping and routing have aligned objectives (*e.g.,* minimize latency) and complete visibility, optimizing their decisions separately taking turns can still lead to globally suboptimal situations. In Fig. 2(a), mapping and routing are locally optimal given each other's decisions, as traffic is served through the least latency paths respecting link capacities. However, the globally optimal traffic allocation is one where all traffic is served by the DC on the right, using both peers.

Bad routing decisions for prefixes with no traffic contribution to a DC can lead to suboptimal equilibria. Consider the scenario in Fig. 2(b), where a mapping initialization (*e.g.,* from geo-proximity) directs all traffic to the DC on the left. No traffic from this user reaches the other DC, so all routing decisions here are equally good. Suppose this DC chooses to route traffic through the 100ms path. Then, this set of mapping and routing decisions are locally optimal to each other—preventing the routing system from exploring the globally optimal 50ms path.

3. MODEL

In this section, we introduce our notation and model for (i) mapping and routing decisions, (ii) the service provider's performance and cost goals, and (iii) optimizing performance using different granularities of information-sharing between mapping and routing decisions. Our notation is summarized in Table 1.

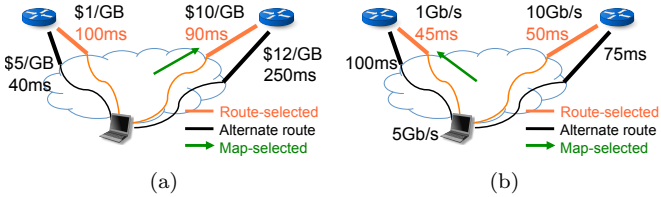


Figure 1: (a) Mapping decision is sub-optimal because of misaligned objectives (b) Mapping decision is sub-optimal because routing does not share link-related information.

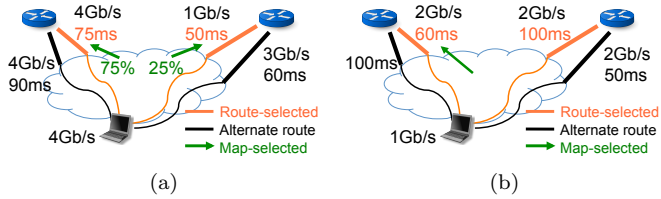


Figure 2: (a) Separate mapping and routing can be inefficient when coupled operational constraints exist; (b) Routing decision is sub-optimal when some clients send no traffic to a DC.

3.1 Cloud Service Provider Model

Network. A cloud service runs in a set I of data centers. Every data center i is connected to the Internet through a set of ISP links, which we denote by J_i. Let C be the set of clients, where each client c is an aggregate of real users (in our experiments, these are IP prefixes). Clients can be directed to different data centers depending on their locations and the current load on the data centers. Such mapping of users to data centers occurs through a set of "mapping nodes", such as authoritative DNS servers or HTTP proxies.

Once a request is serviced at data center i, the egress router there picks one or more ISP-links $j \in J_i$ for responses. The total traffic that a link j can support is limited by its capacity cap_j. Bursting traffic beyond this limit leads to large queueing delays or packet losses—either undesirable for delay-sensitive applications.

Periodic decision-recomputation. Mapping and routing decisions are recomputed periodically, *e.g.,* daily, and then applied to the requests that arrives until the next reoptimization. Hence, any parameters to the optimization *i.e.*, traffic and performance metrics, are averages over the period of optimization.

Replication. We assume that content is fully replicated at all data centers. We believe this is reasonable for read-mostly services which tend to tolerate some stale information in practice. Search, shopping, and social networking applications fall into this category.

Request-servicing at ingress DCs. Some large service providers may bounce requests between data centers through expansive backbone links [3]. However, we assume that user requests are serviced at the first data center which they reach. Further, responses re-enter the Internet through a BGP peer of that data center.

Symbol	Definition
I	Set of data centers (DC), indexed by $i \in I$
J_i	Set of all outgoing links from DC i, indexed by $j \in J_i$
C	Set of client aggregates (prefix), indexed by $c \in C$
vol_c	Total request rate from client c
α_{ic}	Fraction of client c request mapped to DC i
β_{jc}	Fraction of outgoing client c traffic routed on link $j \in J_i$
price_j	Cost ($/request) of routing traffic on link $j \in J_i$
perf_{jc}	Propagation delay between DC i and client c via link j
cap_j	Bandwidth of link $j \in J_i$
$\Phi_j(r_j)$	Performance penalty (congestion cost) as a function of total link load r_j and capacity cap_j

Table 1: Summary of key notation.

Mapping Decisions (choice of servicing-DC). We denote α_{ic} as the proportion of traffic from client c mapped to data center i. We require that $\sum_i \alpha_{ic} = 1$ for all c, where $\alpha_{ic} \in [0, 1]$ for tractability of the optimization. This allows for the possibility that different users in the same client-aggregate c may be served by different data centers at a time. Mapping services such as DNS or HTTP proxies can achieve such flexibility in practice [9, 10], and are accurate enough to be used in multiple commercial deployments, *e.g.,* [6].

While the $\{\alpha_{ic}\}$ denote a choice of data center, the choice of a request's ingress ISP is not explicitly controlled for in our model—we assume that the mapping system only keeps track of one IP address for each (data center, client) pair at a time, namely the shortest-latency path to that client. This allows us to compare multiple mapping schemes with varying levels of routing-system coordination (Table 2 in §3.4). Since ingress (request) traffic tends to be much smaller than responses for typical online services [3], we ignore any link-bandwidth considerations for request traffic.

Routing Decisions (choice of egress-ISP). For every data center i, we denote its response-routing decisions by a set of β_{jc} for all $j \in J_i$, where β_{jc} is the fraction of response-traffic for client c served over link $j \in J_i$. Therefore $\sum_{j:j\in J_i} \beta_{jc} = 1$. Today's BGP routing chooses a single ISP for each IP prefix c, hence routing decisions β_{jc} are integers $\{0, 1\}$. For tractability of optimizations (§3.4) however, we relax this constraint and allow service providers to freely split traffic across links, *i.e.*, $\beta_{jc} \in [0, 1]$. In practice, fractional routing decisions over clients, *e.g.,* IP prefixes, can be realized by hash-based traffic splitting. Since each data center is a stub AS that does not provide transit service, routing changes do not cause BGP convergence issues.

3.2 Performance Goals

User-perceived request latency is a key performance-metric of interest for interactive services – even small increments in this metric can have significant effects on bottomline revenue [11, 1]. User-perceived latency depends on a wide variety of factors, including round-trip latency between users and data centers, request-processing times within data centers, TCP dynamics, and how information is laid out on a user's browser. In this paper, we focus on optimizing the *end-to-end path latency* between users and data centers, which impacts the completion time of short TCP flows [12], much more than metrics like bandwidth and packet loss.

Focus on propagation delay. The end-to-end path latency can be decomposed into three parts: path propagation delay, queueing delays along various bottlenecked links, and transmission delays of packets. Out of these three, we focus on the propagation delay, as it has been shown to largely determine Internet latencies for delay-sensitive applications [13] as opposed to queueing delays. Further, transmission delays for short TCP flows are negligible in the wide-area context.

Average request-delay objective. We employ average request propagation delay (ms/request) as our performance metric. Path propagation delays depend on which data centers handle requests, and which wide-area paths deliver traffic. A service provider obtains the required latencies through active measurements [14] or Internet path performance prediction tools [15]: we use perf_{jc} to denote the propagation delay from data center i to client c, when i picks link $j \in J_i$ to deliver traffic. Then, the performance objective function **perf** is:

$$\textbf{perf} = \left(\sum_{c \in C} \text{vol}_c \sum_{i \in I} \alpha_{ic} \sum_{j \in J_i} \beta_{jc} \text{perf}_{jc} \right) / \sum_c \text{vol}_c,$$

where vol_c is the total request volume from client c.

3.3 Cost Goals

Operational costs are an important consideration for cloud service providers [4, 16]. For tractability, we assume a cost function which is linear on the link traffic workload. In this paper, we focus on ISP-bandwidth costs—although some ISPs employ sophisticated pricing functions (*e.g.,* 95th percentile charging), optimizing a linear cost on every charging interval can reduce the monthly 95th percentile costs [3]. We denote price_j as the cost per request on link $j \in J_i$ of data center i. The cost metric is average cost per request ($/request):

$$\textbf{cost} = \left(\sum_{c \in C} \text{vol}_c \sum_{i \in I} \alpha_{ic} \sum_{j \in J_i} \beta_{jc} \text{price}_j \right) / \sum_c \text{vol}_c$$

3.4 Information-Sharing Granularity

In this section, we introduce our optimization models to study various degrees of information-sharing on the performance goals of a CSP.

Incremental-visibility mapping schemes. For the next two sections (§4, 5) we assume that the mapping and routing systems share the high-level goal of minimizing client latencies. However, the mapping system may not have all the information necessary—*e.g.,* routing decisions—to know *exactly* what latencies will be experienced by client traffic sent to a particular data center. We introduce different degrees of routing-system visibility into mapping, as follows—*cumulatively* adding to prior information as we go along. The resulting mapping schemes are summarized in Table 2, denoting the progression of information-visibility. (For all three mapping schemes, we use a common routing with complete knowledge of mapping decisions, optimizing **perf**.)

A. Shortest latency between each data center and client. A mapping system concerned with minimizing request latencies needs to know at least the closest data center to each client. Scheme *A* maps all traffic to the smallest-latency data center for each client.

B. Aggregate link capacities for each data center. A data center cannot send traffic beyond the aggregate bandwidth of all its ISP-links. A mapping scheme can combine this *static, per-data center* information with client traffic demands to avoid overwhelming data centers. Note that scheme *B* does not use per-link information such as latencies or routing decisions, and directs traffic assuming that the smallest latency path from each data center is used.

C. Per-path latencies, per-link capacities and routing decisions. Mapping scheme *C* has visibility into *per-path information* about the routing system—namely, per-link capacities, per-path latencies, and routing decisions for each prefix. Mapping decisions can potentially be much better with *exact* client-latencies, and capacity considerations of the actual links delivering traffic. This mapping scheme optimizes for the performance objective **perf** (§3.2) as it has all the necessary information. The mapping and routing schemes optimize on top of each other until convergence.

Optimal mapping-routing baseline. To establish a baseline for comparison, we construct a centralized scheme that optimizes both mapping and routing decisions, leveraging complete visibility between the two:

Scheme	Share link caps?	Share all path latencies to DCs?	Share routing decisions?	Objective	Constraints	Variables
A	No	No	No	Average (DC, client) latency	None	$\{\alpha_{ic}\}$
B	Yes	No	No	Average (DC, client) latency	Aggregate DC capacity	$\{\alpha_{ic}\}$
C	Yes	Yes	Yes	Average (path, client) latency	Per-link capacity	$\{\alpha_{ic}\}$
D	-	-	-	Average (path, client) latency	Per-link capacity	$\{\beta_{jc}\}$

Table 2: Summary of mapping schemes ($\{A, B, C\}$) with varying degrees of information-sharing, and the common routing scheme (D). All mapping schemes are aware of the closest latency between (DC, client) pairs, although A only needs to know the closest DC for each client.

GLOBAL

$$\text{minimize} \quad \textbf{perf} \tag{1a}$$

$$\text{subject to} \quad \sum_{c,i:j\in J_i} \text{vol}_c \alpha_{ic} \beta_{jc} \leq cap_j, \ \forall j \tag{1b}$$

$$\sum_i \alpha_{ic} = 1, \ \forall c \tag{1c}$$

$$\sum_{j\in J_i} \beta_{jc} = 1, \ \forall i, c \tag{1d}$$

$$\text{variables} \quad \alpha_{ic} \geq 0, \ \forall i, c, \ \beta_{jc} \geq 0, \ \forall j, c$$

where constraints (1b) ensure that traffic on links do not exceed their capacity. The *global* problem is a linear program on α and β separately, but non-convex when both are variables. As such, it cannot be solved using standard convex optimization techniques. However, Appendix A in our tech report [17] shows that it can be converted into an equivalent LP and solved efficiently.

4. IMPACT ON PROPAGATION DELAY

In this section, we evaluate the set of mapping schemes from Table 2 with a focus to understand the benefits of greater visibility into the routing system. In this section, we perform an offline evaluation, *i.e.*, assume that all schemes have perfect knowledge of the traffic and performance for the next optimization interval. We take up the issue of traffic variability and its impact on performance in §5. We introduce our evaluation setup briefly, followed by results.

4.1 Experiment Setup

We perform all evaluations using trace-based simulations for CoralCDN [18], a caching and content distribution platform running on Planetlab. Our request-level trace collected at 229 Planetlab sites running Coral consists of over 27 million requests and a terabyte of data, corresponding to March 31st, 2011. These requests arrive from about 95000 IP prefixes. For latencies to these prefixes, we use iPlane [15], a system that collects wide-area network statistics to destinations on the Internet from Planetlab vantage points. To correlate the traffic and latency data, we narrow down our client set to 24530 IP prefixes, contributing 38% of the traffic (by requests) of the entire trace. We omit further details in the interest of space, and refer the reader to our tech report (Appendix D, [17]) for complete details. For our experiments, we cluster Planetlab sites in approximately the same metropolitan area and treat them as ISPs connected to the same data center, similar to the approach followed in [2]. Our setup consists of 12 data centers distributed in North America, Europe and Asia with 3-6 ISP connections each.

4.2 Evaluation Results

In this section, we compare the average request propagation delay (**perf**) of various schemes over hourly traffic averages. We show that coarse-grained information-sharing (*i.e.*, scheme B) can provide near-optimal propagation delays—because the smallest latency to a DC is 'representative' of all the path latencies to a client from that DC, when compared to paths from other DCs. We argue that this is due to propagation delays being mostly determined by geographic distance, as opposed to routing choices downstream.

Capacity-awareness is crucial to avoid overwhelming peering ISP links. We find that at all hours of the day, scheme A exceeds the aggregate bandwidth capacity of some data center or the other, so that client traffic can never be allocated within capacity bounds of all its ISP links. The 'popular' data centers every hour are sometimes overwhelmed by more than two times their aggregate capacity. Indeed, the need for capacity-aware traffic management for ISP links and request-mapping has already been highlighted in existing systems [3, 6, 19]. If links are overutilized, queueing delays and packet losses may start dominating performance, severely undermining the significance of propagation delay as a performance metric. For these reasons, henceforth we ignore scheme A from our analysis.

Coarse information-visibility is sufficient for nearly optimal propagation delays. In Fig. 3, we show the hourly variation of the average propagation delay with the capacity-aware schemes, and compare it with

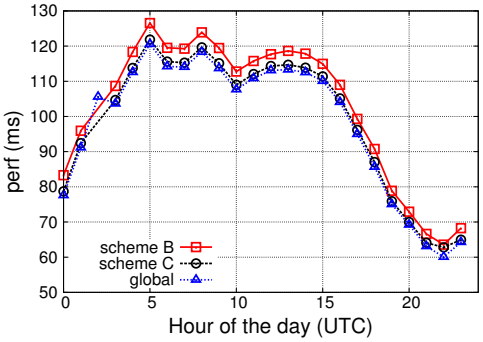


Figure 3: Average request propagation delay on hourly optimizations through the day.

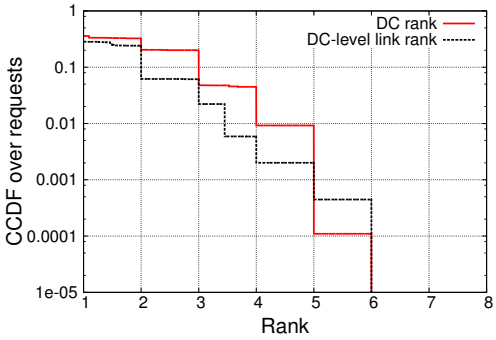


Figure 4: Datacenter and ISP link rank CDFs for scheme B.

the optimal baseline. Throughout the day, we see that aggregate-capacity aware mapping decisions (*i.e.*, scheme B) can already achieve propagation delays very close to optimal – within 10ms difference in this experiment – without knowledge of per-path latencies or routing decisions. As scheme C is initialized with scheme B, we see that it also is very close to optimal, the average gap being less than 2ms. Next, we focus on understanding why these observations hold.

Mapping and routing decisions are nontrivial even under coarse-grained visibility. Our first hypothesis is that the mapping and routing decisions are just implementing a trivial allocation, *i.e.*, using the smallest-latency DC and ISP for most clients. However, Fig. 4 (for hour 0 data) shows that this is *not* the case. We define the *rank* of a data center with respect to a client as its index when the data centers are sorted in increasing order of their shortest latencies to that client. For an ISP link, its rank is its index in the ordering of all ISP links belonging to the same data center. The figure shows that $> 30\%$ of requests are allocated to data centers which are not the closest (rank ≥ 2), while $> 25\%$ of requests are routed through a path which is not the shortest from that data center. This leads us to believe that the shortest latency from a data center to a client is somehow 'representative' of *all* path-latencies from the data center to that client. This is reasonable, since propagation delays are related to geographic distance.

Path-latency variations between peers of the same data center can be high. Our intuition is that if latency variation between peers of a given data center is small, then the shortest latency between the data center and a client can act as a good "proxy" for all path-latencies between that data center and client. Therefore, it can be a sufficient statistic for making good mapping decisions. To quantify this variation, we consider the top four ranked data centers for each client (which cover more than 95% of total traffic by volume), and compute the average difference between all path-latencies from these data centers and the shortest-latency at the same data center.

The CDF of these differences by traffic volume is shown under "all paths" in Fig. 5. We find that there is a sizable traffic volume (*i.e.*, 10%) from clients which have an *average* path-latency difference of > 100ms from the shortest at its data center. This difference is quite significant. However, $> 95\%$ of traffic only traverses the shortest three paths at each data center (Fig. 4), so we perform the averaging only over these paths (curve "shortest 3 paths" in Fig. 5). The average difference reduces significantly, *e.g.,* 95% of requests now have an average difference of < 40ms. However, it is not small enough to explain an optimality gap of < 10ms (Fig. 3).

Latency variations between paths from the same DC are low compared to paths from other DCs. Our hypothesis is that the non-trivial latency variation between peers of the same data center observed above is dwarfed by the variation between ISPs *across* different data centers – so much that the shortest-latency from a data center is an accurate proxy of the latencies from that data center, *when compared to latencies from other data centers*. To quantify this, we compute the difference between intra-data center variation (*i.e.*, average difference from shortest-latency ISP of same DC for top three shortest paths) and inter-data center variation (*i.e.*, average difference of shortest-latencies of top four ranked DCs) for each client.

The CDF by traffic volume is shown in Fig. 6. We see that, *on average*, most requests (*i.e.*, 90%) are from clients whose average path-latency variations *within* data centers are *less* than *between* data centers, *i.e.*, their difference < 0ms. Indeed, 95% of requests are captured by clients whose average intra-data center variation is not more than 10ms higher than the inter-data center variation. We infer that by mapping clients only based on shortest path-latencies, scheme B is not worse-off *on average* by > 10ms for 95% of the total traffic.

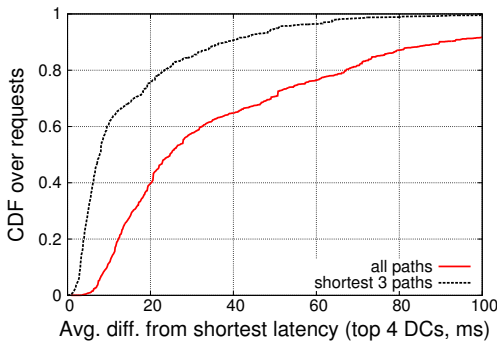


Figure 5: CDF of average latency variation between ISPs within the same DC.

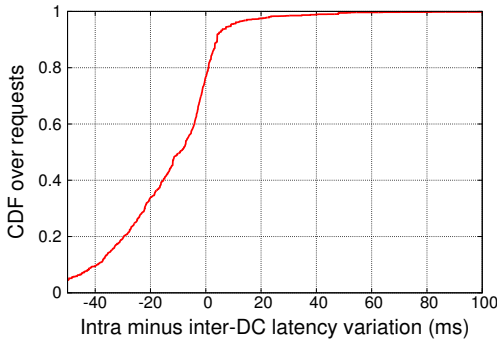


Figure 6: CDF of the difference between intra-DC and inter-DC latency variation metrics.

These latency patterns arise mostly due to geographic distance. We believe that our high-level inferences in this section generalize well, because of a simple physical reason: *geographic proximity is the strongest determinant of propagation delays*, much more so than routing choices of ISPs in practice. A service operator could collect and analyze her own network measurements to see how reasonable a proxy geographic distance is for path latencies, as part of deciding how sophisticated the level of coordination needs to be. Hence, although our specific quantitative results may not be general, we believe the insights are broadly useful.

5. ROBUSTNESS TO TRAFFIC VARIATION

In this section, we compare how different levels of coordination between mapping and routing help guard against traffic variations. These variations can arise in the form of (i) changes in received traffic volumes *between* successive intervals of optimization, *i.e.*, 1 hour in our experiments unless specified otherwise, (ii) request-rate variability *within* an optimization interval, which may arise even if the average arrival rate is unchanged from the previous interval, due to inherent variance in request arrival-rates over shorter timescales. We show that even the most fully-coordinated mapping-routing may not be robust to inter-interval traffic variations (§5.1)—due to high link utilizations caused by traffic burstiness. Further, intra-interval variation can also cause performance disruptions if link utilizations start approaching capacities (§5.2).

5.1 Inter-Interval Traffic Variability

To study inter-interval performance variation, we apply mapping and routing decisions *online*, *i.e.*, compute decisions from traffic measurements over the previous optimization interval, but apply them to traffic in the current interval. We then compare performance against the offline optimal scheme. We have two key observations: (i) propagation delays obtained from optimizing traffic volumes from the *previous* interval are near-optimal for the *current* interval. This is because the median traffic volume change between optimization intervals is quite small. However, (ii) the most bursty traffic can cause very high link utilizations even in fully-coordinated mapping and routing, if links don't have sufficient spare capacity to absorb bursts.

Capacity-aware schemes have nearly optimal average propagation delay when performed online. We find that both scheme B and scheme C in an online scenario are within 10ms of the offline optimal traffic allocations throughout the day (graph omitted due to space constraint). Hence, inter-interval traffic variability does not significantly affect average request propagation delays. Also, fine-grained visibility into routing-system information is not much more advantageous.

Next, we investigate whether the traffic allocations obtained in an online setting are indeed capacity-aware, *i.e.*, within reasonable link utilizations.

Many links are overutilized under both capacity-aware schemes—more visibility is not necessarily beneficial. Fig. 7 depicts the complementary CDF of the utilization of all links across hourly optimizations through the day. (We find similar distributions of link utilizations at any given hour of the trace.) We see that links can be badly overutilized by both schemes B and C. Indeed, at any given hour we find that more than 10% of links are overutilized—leading to significant queueing delays and packet losses. Hence, traffic variability between successive optimization intervals is sufficiently large to make a capacity-respecting traffic allocation from the previous hour into an overutilizing one in the current hour. Next, we understand the traffic variability patterns that cause this observed behavior.

Most traffic volume is quite stable, but there is a long tail of very bursty traffic that leads to high link utilizations. We quantify the traffic variation of a client from the current interval to the next

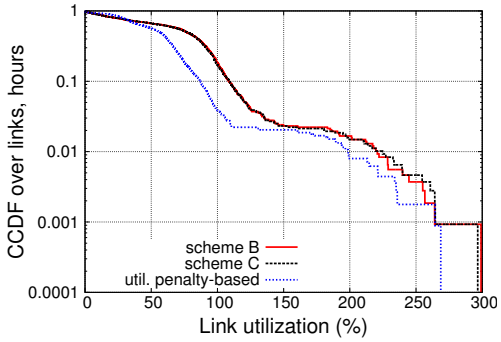


Figure 7: CDF of link utilizations over all links and hours, when mapping and routing decisions are applied in an online fashion every hour.

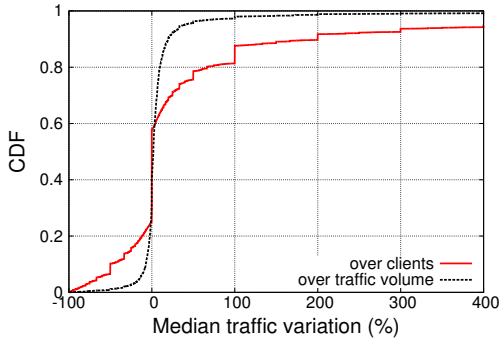


Figure 8: CDF of median hourly traffic variation when aggregated over clients, and over traffic volumes. There are some very bursty clients (curve 'clients'), but most requests are from clients with almost zero median-variation between successive hours (curve 'traffic volume').

as a percentage difference from the current interval's traffic—a 0% change implies a perfect prediction of the next interval, a negative value implies that the current interval's traffic overestimates the next's traffic, and a positive value implies underestimation. We pick every client's median variation from its timeseries (one sample per hour), and plot the CDF over clients and over traffic volumes in Fig. 8. We find that there are a large number of very bursty clients (curve 'over clients'), but a large chunk of traffic volume contribution comes from clients which are quite stable between each hour (curve 'over traffic volume'). However, there is a long tail on the traffic volume distribution—which is the cause of link overutilizations. Indeed, we find that these burstiness patterns in our trace are consistent across multiple reoptimization timescales. Even though the median variation is really good (close to 0%) at all timescales, about 10% of traffic volume in the trace is contributed by some really bursty prefixes at all timescales.

5.2 Intra-Interval Traffic Variability

In this subsection, we look at how schemes with varying degrees of coordination react to *intra-interval* traffic variability. This variability occurs because of the inherently bursty nature of request traffic at multiple timescales [8]—hence, traffic volumes used for mapping and routing decisions are necessarily *averages* of a traffic distribution with a nontrivial variance. As a result, link utilization values resulting from average traffic demands are averages also—and links may be overwhelmed with bursty traffic at shorter timescales, leading to performance disruptions *even if average traffic demand predictions are accurate* for the interval. Naturally, we examine (average) link utilizations over each optimization interval in an offline setting to understand the robustness of coordinated mapping-routing to intra-interval burstiness.

Capacity-aware schemes have high average link utilizations, and fine-grained coordination is not much better. Fig. 9 shows that capacity-constrained schemes tend to push link utilizations up to the maximum allowed capacity for ISP-links corresponding to short-latency paths to clients. Indeed, for both schemes B and C we see that the median link utilization over all optimization intervals and links is at the maximum allowed per link, which is 95% of capacity in this case. As average link utilization approaches capacity, significant queueing delays or packet losses may occur.

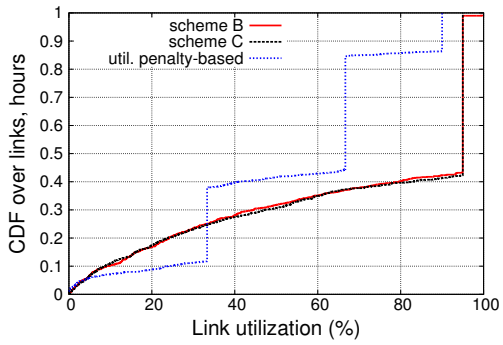


Figure 9: CDF of link utilizations over all links and hours in an offline setting.

In the next section, we explore how to mitigate the ill-effects of traffic burstiness that occur even in the most completely coordinated mapping and routing schemes.

6. ROBUSTNESS WITH SPARE CAPACITY

Our observations from §5 suggest that traffic burstiness across optimization intervals can cause undesirable link overutilization, as can transient traffic bursts—even when the mapping system is completely aware of routing-

system information, and vice-versa. In this section, we explore a simple technique to achieve robustness to traffic burstiness: *enforcing some bandwidth headroom* while computing mapping and routing decisions. By leaving some unused "slack" capacity in each link while computing mapping and routing decisions, traffic burstiness can be accommodated much better.

The simplicity of this idea makes it appealing, as opposed to other techniques such as AS-level traffic aggregation [20], which makes it challenging to distinguish clients within an AS by performance, or employing more sophisticated estimators for traffic arrivals—which may still prove ineffective against transient bursts.

Below, we explore the question of setting a fixed capacity "reserve" for absorbing bursts—but show that we need to navigate a tricky tradeoff between the reduction in the number of highly-utilized links and latency increase which occurs due to traffic taking larger-latency paths. Then, in §6.1 we introduce the idea of penalizing high utilizations naturally in the objective function to achieve better robustness, and show in §6.2 that this also provides optimality guarantees.

How much slack capacity should be used? We ask what is a "good" fraction of capacity to reserve to absorb bursty traffic. The answer depends on the volumes of the traffic bursts and link capacities, in practice. If the fraction of spare capacity reserved is too low, then it might not be sufficient to absorb incoming traffic bursts. However, if the fraction is too high, then in the event that the link is not bursted, traffic is allocated to longer-latency paths even when shorter latency paths are available—leading to unutilized capacity and larger request latencies.

Indeed, we find that as the fraction of capacity reserved for bursts increases, the fraction of overutilized links through the entire trace reduces—from 24% at no reservations to 4% at 30% capacity reservation. However, the average propagation delay increases compared to what could have been achieved with the full link capacities, since traffic is now forced to use longer-latency paths *even if the reserved capacity is not bursted.* The average propagation delay gap goes up from 5ms at no reservations to 22ms at 30% reservations for scheme C, and 10ms and 26ms for scheme B respectively.

6.1 Dynamic Utilization Penalties

Balancing the tradeoff between average request-delay and link-overutilization while picking a fixed capacity-reservation can be tricky. Instead, we introduce an alternate approach: *penalize high link utilizations* in the objective function of the optimization—with higher penalties for higher utilizations. Intuitively, a penalty on link utilizations dynamically reduces link loads by increasing the value of the objective function, but permits higher utilizations in cases where such reduction would come at the cost of a large performance loss, *i.e.*, increase in propagation delay.

The utilization penalty. Instead of using hard capacity limits in our optimization models, we "relax" the link capacity constraints by introducing a penalty function. In particular, we utilize a piece-wise linear function $\Phi_j(\cdot)$ (Appendix C, [17]) to capture such penalties due to high link utilization, and is often used in ISP traffic engineering [21]. We introduce a refined objective function, namely $\mathbf{perf_\Phi}$, which replaces the objectives from scheme C and scheme D by

$$\begin{aligned}\mathbf{perf_\Phi}(\alpha,\beta) &= \sum_{i,j\in J_i,c} \alpha_{ic}\beta_{jc}\mathrm{vol}_c\mathrm{perf}_{jc} / \sum_c \mathrm{vol}_c \\ &\quad + \sum_{i,j\in J_i} \Phi_j\left(\sum_c \alpha_{ic}\beta_{jc}\mathrm{vol}_c\right) / |J| \quad (2)\end{aligned}$$

where $J = \bigcup_i J_i$, the set of all ISP links. $\mathbf{perf_\Phi}$ can be viewed as an average request-latency with a "link congestion" consideration. The link capacity constraints are then removed from both problems.

The utilization-penalty based scheme improves robustness to traffic variations over both the other schemes. We evaluate the robustness of the utilization-penalty based mapping and routing schemes to traffic variability in both online and offline scenarios. In the online case (Fig. 7), we find that utilization penalties can lower the fraction of overutilized links from 20% to $< 5\%$. However, the distribution continues to be long-tailed—we believe is due to very high worst-case traffic variability. In the offline case (Fig. 9), link utilizations are in general lower than the other schemes, and distributed gradually between 0 and 90%, the step function resulting from piecewise-linearity of the penalty function. The generally lower utilizations help reduce the effect of short-term traffic bursts.

In all our runs we find that the average propagation delays of the utilization-penalty based scheme are within 15ms of the *global* propagation delay optimum.

6.2 Optimality

The utilization penalty function provides a serendipitous benefit in addition to being more robust—alternate mapping-routing optimizations can be made to converge provably to the global optimum of the new objective function $\mathbf{perf_\Phi}$ over the combined feasible space of the mapping and routing variables, bypassing the difficulties in §2. We provide our key optimality theorem and intuition here, and refer the reader to our tech report (Appendix B, [17]) for the complete proof.

Optimality theory. We denote the transformed mapping and routing optimizations with objective $\mathbf{perf_\Phi}$ and link capacity constraints removed, as C_Φ and D_Φ

respectively. We also define the transformed baseline $\mathbf{GLOBAL_{\Phi}}$ in a similar fashion, replacing the objective function with $\mathbf{perf_{\Phi}}$ and removing link capacity constraints. Intuitively, we get around the 'coupled operational constraint' problem in §2 by converting hard capacity constraints into 'soft' objective function penalties. We deal with bad routing decisions in the absence of traffic (§2) by adopting the refinement from [7]. We term a mapping or routing optimization with this refinement as an *optimal projection* (Def. 2). Our key optimality result is as follows.

Theorem 1. *Optimal projections of C_Φ and D_Φ converge to an optimal solution of* $\mathbf{GLOBAL_{\Phi}}$.

Due to space constraint, we only provide the proof sketch here and refer the reader to our tech report (Appendix B, [17]) for the complete proof. The proof proceeds in three steps. We first show that there exists an equilibrium point (α^*, β^*), such that α^* is a *best response* to β^*, and vice versa. We define the best response as an *optimal projection* (Def. 2), and the resulting equilibrium as a *Nash equilibrium* (Def. 3). We further prove that (α^*, β^*) is also an optimal solution to $\mathbf{GLOBAL_{\Phi}}$, extending previous results [7] to models with linear mapping constraints. We finally show that iterative optimal projections lead to a Nash equilibrium point, completing the proof.

We introduce the formalities to assist our proof. First, we need a way to capture the marginal cost f_{ijc} of serving client c from data center i over link $j \in J_i$:

Definition 1. *Let the metric f_{ijc} be defined as*

$$f_{ijc}(\alpha_{ic}, \beta_{jc}) = \mathrm{vol}_c \cdot \mathrm{perf}_{jc} / \sum_{c'} \mathrm{vol}_{c'} + \mathrm{vol}_c \Phi'_j \left(\sum_c \alpha_{ic}\beta_{jc}\mathrm{vol}_c \right) / |J|$$

We next define the optimal projection of mapping and routing strategies, respectively:

Definition 2. *(i) Given fixed routing decisions β, a set of mapping decisions α^* is called an optimal projection onto mapping space if α^* is an optimal solution to C_Φ. (ii) Given fixed mapping decisions α, a set of routing decisions β^* is called an optimal projection onto routing space if, $\forall i, c$, we have $f_{ijc}(\alpha_{ic}, \beta^*_{jc}) \leq f_{ij'c}(\alpha_{ic}, \beta^*_{j'c})$ for all $j, j' \in J_i$ such that $\beta^*_{jc} > 0$.*

We next define the notion of *Nash equilibrium*.

Definition 3. *A set of mapping and routing decisions (α^*, β^*) is called a Nash equilibrium if α^* is an optimal projection given β^*, and β^* is an optimal projection given α^*.*

We state the results which string the proof together:

Theorem 2. *There exists a Nash equilibrium.*

Theorem 3. *Let (α^*, β^*) be a Nash equilibrium. Then $\{\alpha^*, \beta^*\}$ is an optimal solution to* $\mathbf{GLOBAL_{\Phi}}$.

Theorem 4. *Optimal projections of C_Φ and D_Φ converges to a Nash equilibrium.*

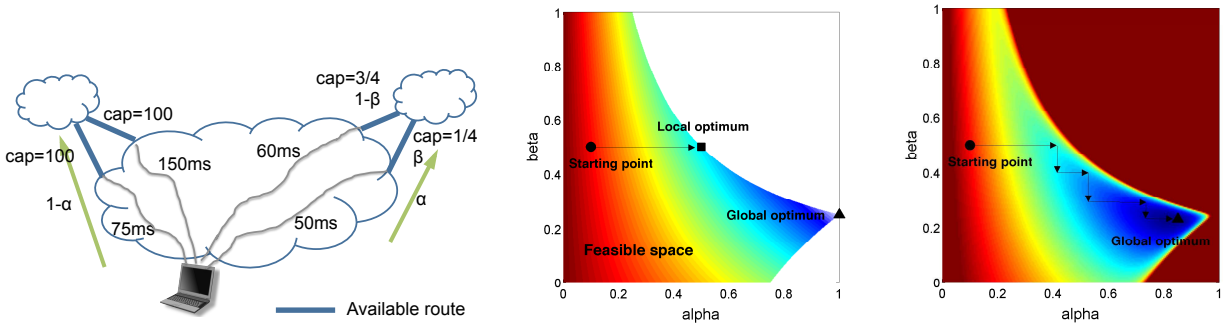
Visualization. Consider the toy example illustrated in Figure 10(a). Let α and β be the mapping and routing decisions respectively. Suppose the client has one unit of demand, and the goal of both mapping and routing is to minimize latency. The optimal performance is achieved when all traffic is mapped to the DC at the right, with routing set such that no link capacity is violated, *i.e.*, $\alpha = 1, \beta = 1/4$. However, by separate optimizations, both parties can easily get stuck in a local optimum. For instance, consider $\alpha = 1/2, \beta = 1/2$, which are valid mapping and routing settings. Mapping is optimal as increasing α overshoots the link capacity. Routing is also optimal because routing 1/2 of the traffic on the 50ms link makes it fully utilized.

Figure 10(b) visualizes how the local optimum is reached from an initial starting point. The color-shaded region represents the feasible decision space. The curvy boundaries reflect the capacity constraints of the 50ms and 60ms links. The color coding represents the objective value, *i.e.*, latency, where red means high values and blue, low. All points on these boundaries except the rightmost one are local optima with higher latencies than the global optimum. In general, the feasible space determined by link capacity constraints is non-convex.

Figure 10(b) suggests that local equilibria only exist on the boundaries that are implied by the link capacity constraint. Hence, we focus on this constraint—and prove that removing it removes the suboptimality from alternate optimizations. Indeed, when we remove the capacity constraint and introduce the penalty into the objective function, we arrive at the situation in Fig. 10(c). The whole decision space is now feasible, but the violation of the link capacities will imply a high objective (shown by dark red color). Further, there is a gradient of objective values near the (previous) capacity boundary, transitioning from low to very high values of the objective function.

Indeed, this objective gradient allows alternate mapping-routing optimizations to converge to the global optimum as shown in Fig. 10(c). Note that the global optima shown in Figures 10(b) and 10(c) are slightly different, as the congestion cost is not considered in the original formulation. As noted previously in §6.1, the difference in optimal propagation delays is within 15ms in all our experiments.

7. IMPACT ON COST



(a) Example network (b) GLOBAL (c) GLOBAL$_\Phi$

Figure 10: An example of local optima with separate mapping-routing optimization.

We evaluate the impact of per-link information visibility for costs. We introduce mapping schemes analogous to schemes B and C—namely $B.1$ and $C.1$—which use *average* costs of all links in a DC and per-link bandwidth costs, respectively. We also introduce the centralized scheme which optimizes for the average cost metric (§3), *global*.1. (The costs per unit bandwidth for links are derived from a cloud bandwidth pricing plan [22]; we refer the reader to our tech report for details [17]).

Visibility into per-link costs provides a significant advantage. We find that scheme $B.1$ can incur significantly worse costs (25% higher on average) than scheme $C.1$. This is reasonable, as prior works have noted that the bandwidth-cost diversity between ISPs can be quite large [3, 23]. We find that $C.1$ achieves near-optimal costs most hours of the day, with a few outliers (due to situations like in §2).

8. GENERALIZABILITY

A natural concern that may arise from the empirical observations in this paper is whether they generalize beyond the specific CDN traces that we have used. We argue that even though specific quantitative results from this work may not hold in general, our insights and methodologies are still broadly useful.

Propagation delay comparisons. A service operator can easily determine through the described method and his own network measurements whether propagation delay is indeed the key determinant of latency variation patterns—thereby deciding on a suitable level of coordination. Indeed, some publicly known state-of-the-art mapping systems, *e.g.,* [19] default to geographically closest replicas with some load information.

Robustness to traffic burstiness. Burstiness of web traffic at various timescales has been observed in multiple environments [8, 20, 24]. Clearly, the proportion of highly utilized or overutilized links is a network bandwidth and workload-dependent property. However, the general traffic burstiness characteristics and their implications for link load are in agreement with observations from prior works on traffic engineering which attempt to reduce link utilizations, *e.g.,* [21].

Cost comparisons. The large diversity of ISP bandwidth costs has been observed in prior work [3, 23], motivating schemes that attempt to minimize costs by utilizing low-cost links as opposed to high-cost ones. A network operator can easily determine if this diversity plays a key role in her network costs through the evaluations as described.

Robust, optimal coordination scheme. Our optimality result is independent of any specifics of our dataset, and can provide useful performance-predictability for service operators.

9. RELATED WORK

Joint control. Recently, there has been work on joint control and interaction between two parties for traffic engineering. DiPalatino *et al.* [7] model a game in which two players have independent control over routing and server selection, and determine when a social planner might find optimal utilities. Jiang *et al.* [25] propose cooperative server selection and traffic engineering between network and content providers who have conflicting objectives. We propose an extension of the optimality result (Appendix B, [17]) incorporating practical considerations such as DC-level load balancing and capacity constraints. In addition, we evaluate multiple formulations corresponding to varying degrees of visibility.

Wide-area traffic engineering. WhyHigh [6] diagnoses latency problems with closest-replica mapping, categorizing their causes into peering, capacities, and ISP traffic engineering. Our work is complementary to this approach, as it embraces the possibility that

closest-replica mapping does not always offer best performance, and focusses on the benefits of coordination *given* the current ISP peers and link capacities.

Entact [3] proposes to optimize traffic engineering within Microsoft's backbone across all upstream ISPs, when requests enter through the closest ingress point. In contrast, we consider the impact of the *choice of ingress point i.e.*, mapping, which is crucial for CSPs without a backbone network. Hence, we also consider the functional separation between mapping and routing.

Scalable networked services. Volley [26] considers the problem of data placement across the wide-area, but we assume that content is read-mostly and fully replicated. Goldenberg *et al.* [23] studied multi-homed traffic engineering for stub autonomous systems to optimize cost and performance. DONAR [9] is a decentralized mapping service that allows customized client policies and offers better performance to clients. In contrast to these latter two works, we study a set of distributed solutions coordinating mapping *and* routing.

Network performance measurement. There is a rich body of research on network path performance estimation, which is complementary to our study, for determining the $\text{perf}_{\mathbf{jc}}$ metric. Virtual coordinate systems *e.g.,* [27] estimate latency based on synthetic coordinates. Others focus on the reducing the overheads for measuring IP address space, *e.g.,* [28]. Our study employs a path performance prediction service [15] that builds on segments of known Internet paths.

10. CONCLUSION

We present the problem of coordinating data center selection and response-routing for online services, which can allow service providers to offer better performance to users at lower cost. We provide optimization formulations and use real CDN traffic traces to study varying levels of information visibility, and find that (i) sharing coarse-grained information is sufficient to achieve good latencies, but (ii) fine-grained information is necessary for minimal costs. Further, (iii) even systems with full information visibility may lack robustness to traffic variations. Finally, we propose a robust coordination technique which retains administrative separation between the two systems, and is also provably optimal.

11. REFERENCES

[1] J. Hamilton, "The cost of latency." `http://perspectives.mvdirona.com/2009/10/31/TheCostOfLatency.aspx`.

[2] A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman, "A measurement-based analysis of multihoming," in *Proc. ACM SIGCOMM*, 2003.

[3] Z. Zhang, M. Zhang, A. Greenberg, Y. C. Hu, R. Mahajan, and B. Christian, "Optimizing cost and performance in online service provider networks," in *Proc. NSDI*, 2010.

[4] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," in *Proc. ACM SIGCOMM*, 2009.

[5] Y. Chen, S. Jain, V. K. Adhikari, and Z.-L. Zhang, "Characterizing roles of front-end servers in end-to-end performance of dynamic content distribution," in *Proc. Internet Measurement Conference*, 2011.

[6] R. Krishnan, H. V. Madhyastha, S. Srinivasan, S. Jain, A. Krishnamurthy, T. Anderson, and J. Gao, "Moving beyond end-to-end path information to optimize cdn performance," in *IMC*, 2009.

[7] D. DiPalantino and R. Johari, "Traffic engineering vs. content distribution: A game theoretic perspective," in *Proc. IEEE INFOCOM*, 2009.

[8] A. Feldmann, A. C. Gilbert, and W. Willinger, "Data networks as cascades: Investigating the multifractal nature of internet wan traffic," in *SIGCOMM*, 1998.

[9] P. Wendell, J. W. Jiang, M. J. Freedman, and J. Rexford, "DONAR: Decentralized server selection for cloud services," in *Proc. ACM SIGCOMM*, 2010.

[10] "Client subnet in DNS requests," 2012. IETF draft.

[11] R. Kohavi, R. M. Henne, and D. Sommerfield, "Practical guide to controlled experiments on the web: Listen to your customers not to the hippo," in *Proc. ACM SIGKDD*, 2007.

[12] N. Cardwell, S. Savage, and T. Anderson, "Modeling tcp latency," in *Proc. IEEE INFOCOM*, 2000.

[13] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and C. Diot, "Packet-level traffic measurements from the Sprint IP backbone," *IEEE Network*, 2003.

[14] M. Szymaniak, D. Presotto, G. Pierre, and M. van Steen, "Practical large-scale latency estimation," *Computer Networks*, vol. 52, pp. 1343–1364, May 2008.

[15] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iPlane: An information plane for distributed services," in *Proc. OSDI*, Nov. 2006.

[16] J. Hamilton, "Overall data center costs." `http://perspectives.mvdirona.com/2010/09/18/OverallDataCenterCosts.aspx`.

[17] S. Narayana, J. W. Jiang, J. Rexford, and M. Chiang, "Technical report." `http://www.cs.princeton.edu/~narayana/joint-maproute.html`.

[18] M. J. Freedman, E. Freudenthal, and D. Mazières, "Democratizing content publication with Coral," in *Proc. NSDI*, 2004.

[19] Y. Zhu, B. Helsley, J. Rexford, A. Siganporia, and S. Srinivasan, "LatLong: Diagnosing wide-area latency changes for CDNs." In submission, `http://www.cs.princeton.edu/~jrex/papers/latlong.pdf`.

[20] N. Feamster, J. Borkenhagen, and J. Rexford, "Guidelines for interdomain traffic engineering," *CCR*, 2003.

[21] B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional IP routing protocols," *IEEE Communication Magazine*, vol. 40, pp. 118 – 124, Oct 2002.

[22] Amazon EC2 Pricing. http://aws.amazon.com/ec2/pricing/.

[23] D. K. Goldenberg, L. Qiu, H. Xie, Y. R. Yang, and Y. Zhang, "Optimizing cost and performance for multihoming," in *Proc. ACM SIGCOMM*, 2004.

[24] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Trans. Networking*.

[25] J. W. Jiang, R. Zhang-Shen, J. Rexford, and M. Chiang, "Cooperative content distribution and traffic engineering in an ISP network," in *Proc. ACM SIGMETRICS*, 2009.

[26] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, A. Wolman, and H. Bhogan, "Volley: automated data placement for geo-distributed cloud services," in *Proc. NSDI*, 2010.

[27] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system," in *Proc. ACM SIGCOMM*, 2004.

[28] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "IDMaps: A global Internet host distance estimation service," *IEEE/ACM Trans. Networking*, 2001.

[29] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing router buffers," in *SIGCOMM*, 2004.
[30] A. Bavier, M. Bowman, B. Chun, D. Culler, S. Karlin, S. Muir, L. Peterson, T. Roscoe, T. Spalink, and M. Wawrzoniak, "Operating system support for planetary-scale network services," in *NSDI*, 2004.
[31] The RouteViews Project. http://www.routeviews.org/.

Symbol	Definition
J	Set of all outgoing ISP links, *i.e.*, $\bigcup_i J_i$
X_{ijc}	Fraction of client c traffic routed through link $j \in J_i$ That is, $\sum_{i,j\in J_i} X_{ijc} = 1$ for all c.
K	Cost-performance tradeoff parameter
P_i	Proportion of total traffic directed to DC i
w_i	Target traffic split ratio of DC i
ϵ_i	Tolerance of deviation from target split ratio
B_i	Request-rate cap placed on DC i
f_{ijc}	Marginal cost function for link $j \in J_i$, client c
g_c	Fraction of traffic contributed by client c
$\lambda_{i1}, \lambda_{i2}$	Dual optimal variables for the $GLOBAL$ problem
μ_{i1}, μ_{i2}	Dual optimal variables for the $MAPPING$ problem

Table 3: Additional notation introduced in this appendix for model generalization and proof.

APPENDIX

In this appendix, we extend the mapping and routing with complete information exchange (*i.e.*, schemes C and D) with two considerations: (i) joint cost and performance optimization, and (ii) data center load management. These extended formulations are referred to as **MAPPING** and **ROUTING** respectively. We also rewrite the baseline problem **GLOBAL**. These extensions subsume formulations C and D, and the **GLOBAL** formulation provided earlier in the paper. Consistent with earlier notation, we denote formulations with link utilization penalties instead of link capacity constraints using a Φ in the subscript: namely, $\mathbf{MAPPING}_{\mathbf{\Phi}}$, $\mathbf{ROUTING}_{\mathbf{\Phi}}$ and $\mathbf{GLOBAL}_{\mathbf{\Phi}}$ respectively.

The appendixes are organized as follows. Appendix A shows that the extended global problem, which is non-convex, is equivalent to a centralized *convex* optimization problem. Appendix B proves the optimality of alternating mapping and routing optimizations (with extended formulations), when link utilization penalties are used. Appendix C shows the form of this link utilization penalty function Φ. Appendix D describes the data setup for the evaluations. The additional notation introduced in the appendixes is described in Table 3.

Model Extensions

Jointly optimizing performance and cost. The goals of maximizing performance and minimizing costs are often at odds. "Good" links that offer lower latencies are usually priced higher, and over-utilizing low-cost links will degrade performance due to increased congestion. Therefore, CSPs need to customize the cost-performance tradeoff. To strike a balance between cost and performance, we introduce a weight factor K that reflects the amount of additional costs ($/request) that a CSP is willing to pay for one unit of performance improvement (ms/request). That is, the CSP's goal is to minimize the objective function $\mathbf{cost} + K \cdot \mathbf{perf}$. Note that the sub-cases of performance-only and cost-only optimizations are obtained by appropriate settings of K, namely $K = 0$ and $K = \infty$ respectively.

Data center load management. For additional flexibility, we allow CSPs to express their preferences of a traffic split ratio w_i for each data center i, with a tolerance ϵ_i of deviation for other purposes like performance and cost. This allows CSPs to balance workload between data centers proportional to their server resources, or intentionally shift traffic to under-utilized data centers due to unattractive network locations, *e.g.,* large RTTs to reach the majority of clients. CSPs usually need to place a cap on the total requests a DC receives according to its computing power, *e.g.,* proportional to the number of servers in that DC. We let CSPs specify such a request-rate cap, *e.g.,* B_i, for every data center i as its capacity limit. Note that appropriate settings of w_i, ϵ_i and B_i allow us to easily reflect formulations without these constraints, *e.g.,* setting $\epsilon_i = 1.0, w_i \in [0, 1]$, and $B_i > \sum_c \text{vol}_c$.

With these extensions, the mapping, routing and global baseline formulations are written down as follows.

$$\begin{aligned}
&\mathbf{MAPPING}(\beta)\\
&\text{minimize} && \mathbf{cost} + K \cdot \mathbf{perf} && (3a)\\
&\text{subject to} && \left| \frac{\sum_c \alpha_{ic}\text{vol}_c}{\sum_c \text{vol}_c} - w_i \right| \le \epsilon_i,\ \forall i && (3b)\\
& && \sum_c \alpha_{ic}\text{vol}_c \le B_i,\ \forall i && (3c)\\
& && \sum_{c,i:j\in J_i} \text{vol}_c \alpha_{ic}\beta_{jc} \le cap_j,\ \forall j && (3d)\\
& && \sum_i \alpha_{ic} = 1,\ \forall c, && (3e)\\
&\text{variables} && \alpha_{ic} \ge 0,\ \forall\ i, c
\end{aligned}$$

where constraints (3b) and (3c) represent data center-level load management constraints, namely load splitting and load capping. Constraints (3d) represent per-link bandwidth capacities.

$$\begin{aligned}
&\mathbf{ROUTING}(\alpha)\\
&\text{minimize} && \mathbf{cost} + K \cdot \mathbf{perf} && (4a)\\
&\text{subject to} && \sum_j \beta_{jc} = 1,\ \forall i, c && (4b)\\
& && \sum_{c,i:j\in J_i} \text{vol}_c \alpha_{ic}\beta_{jc} \le cap_j,\ \forall j && (4c)\\
&\text{variables} && \beta_{jc} \ge 0,\ \forall\ i, j, c
\end{aligned}$$

GLOBAL

$$\text{minimize} \quad \mathbf{cost} + K \cdot \mathbf{perf} \tag{5a}$$

$$\text{subject to} \quad \sum_{c,i:j\in J_i} \text{vol}_c \alpha_{ic} \beta_{jc} \le cap_j, \ \forall j \tag{5b}$$

$$\left| \frac{\sum_c \text{vol}_c \alpha_{ic}}{\sum_{c'} \text{vol}_{c'}} - w_i \right| \le \epsilon_i, \ \forall i \tag{5c}$$

$$\sum_c \text{vol}_c \alpha_{ic} \le B_i, \ \forall i \tag{5d}$$

$$\sum_i \alpha_{ic} = 1, \ \forall c \tag{5e}$$

$$\sum_{j\in J_i} \beta_{jc} = 1, \ \forall i, c \tag{5f}$$

$$\text{variables} \quad \alpha_{ic} \ge 0, \ \forall i, c, \ \beta_{jc} \ge 0, \ \forall j, c$$

A. A CENTRALIZED APPROACH TO THE JOINT PROBLEM

We present an equivalent convex formulation of the joint mapping and routing optimization problem that accepts an efficient centralized solution. As written above (5), **GLOBAL** is a non-convex optimization problem on variables $(\alpha_{ic}, \beta_{jc})$, and hence cannot be solved using standard convex programming techniques. It is not clear whether local optima exist and how bad they are relative to the global optimum. However, we show that (5) can be translated into an equivalent convex optimization problem by introducing a new set of variables X_{ijc}, which represents the fraction of client c traffic that is mapped to data center i through link j. There is a direct connection between the two sets of variables: $X_{ijc} = \alpha_{ic} \cdot \beta_{jc}, \ \forall (i, j, c)$ We can rewrite (5) into the following convex program:

GLOBAL$_X$

$$\text{minimize} \quad \frac{\sum_{ijc} X_{ijc} \text{vol}_c \left(\text{price}_j + K \cdot \text{perf}_{jc}\right)}{\sum_{c'} \text{vol}_{c'}} \tag{6a}$$

$$\text{subject to} \quad \left| \frac{\sum_{c,j\in J_i} \text{vol}_c X_{ijc}}{\sum_{c'} \text{vol}_{c'}} - w_i \right| \le \epsilon_i, \ \forall i \tag{6b}$$

$$\sum_{c,j\in J_i} \text{vol}_c X_{ijc} \le B_i, \ \forall i \tag{6c}$$

$$\sum_{c,i:j\in J_i} \text{vol}_c X_{ijc} \le cap_j, \ \forall j \tag{6d}$$

$$\sum_{ij} X_{ijc} = 1, \ \forall c \tag{6e}$$

$$\text{variables} \quad X_{ijc} \ge 0 \ \forall \ i, j, c$$

As we show in the theorem below, the optimal solutions of **GLOBAL** and **GLOBAL$_X$** are related, and the two problems have the same optimal objective.

Theorem 5. **GLOBAL** *and* **GLOBAL$_X$** *are equivalent. They have the same optimal objective values. Further, their optimal solutions can be derived from each other.*

PROOF. First, the optimal value of **GLOBAL$_X$** is a lower-bound on that of **GLOBAL**. Consider any feasible solution $\{\alpha_{ic}, \beta_{jc}\}$ of (5). We construct $X_{ijc} = \alpha_{ic} \cdot \beta_{jc}$, which is also feasible for problem (6). It can be readily verified that two solutions achieve the same objective value. Second, the optimal value of **GLOBAL** is a lower-bound on that of **GLOBAL$_X$**. Consider any feasible solution $\{X_{ijc}\}$ of (6). We construct $\alpha_{ic} = \sum_{j\in J_i} X_{ijc}$, $\beta_{jc} = X_{ijc}/\alpha_{ic}$ if $\alpha_{ic} > 0$ and $\beta_{jc} = 1/|J_i|$ otherwise. It can be verified that $\{\alpha_{ic}, \beta_{jc}\}$ is feasible for **GLOBAL**, and achieves the same objective value as **GLOBAL$_X$**. We establish a one-to-one mapping between variables of (5) and (6), and hence they obtain the same optimal objectives and solutions. ∎

It is also easily checked that the equivalence continues to hold between the two new optimization problems formed when **GLOBAL** and **GLOBAL$_\mathbf{X}$** are refined by including link utilization penalties in the objective instead of hard capacity constraints (5b) and (6d).

B. OPTIMALITY OF DISTRIBUTED MAPPING AND ROUTING

We present the proof of optimality of alternating mapping and routing optimizations with link utilization penalties—referred to as **MAPPING$_\mathbf{\Phi}$** and **ROUTING$_\mathbf{\Phi}$**. These are obtained from (3) and (4) above respectively by replacing the objective function by $\mathbf{cost} + K \cdot \mathbf{perf_\Phi}$, and removing the link capacity constraints (3d) and (4c). At a high level, the algorithm proceeds as follows: given routing decisions β as input, mapping nodes solve (3) and optimize over α, and given mapping decisions α as input, edge routers solve (4) and optimize over β. The two steps are carried out iteratively until solutions converge. We allow mapping and routing problems to be solved at different time-scales, with the only requirement that (3) does not start until (4) is fully solved, and vice versa. This is easily ensured by having each component wait to receive inputs from the other before solving its own local optimization problem.

However, we showed in Figure 2(b) that in general the alternate projection algorithm may still lead to suboptimal equilibria, when some clients send no traffic to a data center. To overcome this issue, we introduce a refinement to routing called *optimal projection*, in addition to the optimality of (4). We soon prove the optimality of such a refined routing strategy. In practical computation of routing decisions, we introduce an approximation to (4) by incrementing an infinitesimally small demand to a client-server pair with zero traffic, *i.e.*, $\alpha_{ic} \leftarrow \delta$ if $\alpha_{ic} = 0$, where δ is a small positive constant [7]. This approximation is often used to ease

the computation such that standard optimization techniques can be applied. However, clients do not need to send real traffic, making this approach practically appealing.

We redefine the metric f_{ijc} (previously Def. 1) and optimal projection (previously Def. 2) corresponding to the extended optimization formulations. Note that the notion of Nash equilibrium (Def. 3) is reinterpreted through the modified definition of optimal projection.

Definition 4. *Let the metric f_{ijc} be redefined as*

$$f_{ijc}(\alpha_{ic}, \beta_{jc}) = \mathrm{vol}_c \left(\mathrm{price}_j + K \cdot \mathrm{perf}_{jc}\right) / \sum_{c'} \mathrm{vol}_{c'} + \mathrm{vol}_c \Phi'_j \left(\sum_c \alpha_{ic}\beta_{jc}\mathrm{vol}_c \right) / |J|$$

Definition 5. *(i) Given fixed routing decisions β, a set of mapping decisions α^* is called an optimal projection onto mapping space if α^* is an optimal solution to* $\mathbf{MAPPING_{\Phi}}$. *(ii) Given fixed mapping decisions α, a set of routing decisions β^* is called an optimal projection onto routing space if, $\forall i, c$, we have $f_{ijc}(\alpha_{ic}, \beta^*_{jc}) \le f_{ij'c}(\alpha_{ic}, \beta^*_{j'c})$ for all $j, j' \in J_i$ such that $\beta^*_{jc} > 0$.*

We prove the results stringing the proof together in the subsections below. Even though we follow the definitions and proofs of convergence and optimality in [7], we cannot directly apply their results because of the presence of data center load constraints (3b) and (3c).

B.1 Optimal Projection implies Optimality

Lemma 1. *If β^* is an optimal projection given α, then β^* is also an optimal solution to* $\mathbf{ROUTING_{\Phi}}$.

Proof. To show that β^* is an optimal solution to the routing problem (*i.e.*, (4) with refined objective), we check the KKT conditions for (4). These conditions hold if and only if $\alpha_{ic} f_{ijc} \le \alpha_{ic} f_{ij'c}, \forall j, j' \in J_i$ and $\beta_{jc} > 0$. This is true because when $\alpha_{ic} > 0$, the definition of optimal projection implies the above condition. When $\alpha_{ic} = 0$, the optimality condition holds too. ∎

It is easy to check that an optimal solution to (4) is not necessarily an optimal projection.

B.2 Existence of Nash Equilibrium

Theorem 6. *There exists a Nash equilibrium.*

Proof. We show this by construction. Let X^* be an optimal solution to $\mathbf{GLOBAL}_X$ with refined objective function $\mathbf{cost} + K \cdot \mathbf{perf_{\Phi}}$ and without capacity constraint (6d). We construct a set of mapping and routing decisions as follows: $\alpha^*_{ic} = \sum_{j \in J_i} X^*_{ijc}$, $\beta^*_{jc} = X^*_{ijc}/\alpha^*_{ic}$ if $\alpha^*_{ic} > 0$. If $\alpha^*_{ic} = 0$, we set $\beta^*_{j'c} = 1$ for some $j' = \mathrm{argmin}_{j \in J_i} f_{ijc}$, and set $\beta^*_{jc} = 0$ for $j \in J_i \backslash \{j'\}$. By Theorem 5, it is easy to check that (α^*, β^*) is also an optimal solution to $\mathbf{GLOBAL_{\Phi}}$. Therefore, α^* must be an optimal solution to $\mathbf{MAPPING_{\Phi}}(\beta^*)$, as otherwise we can find a better α to improve the global objective function. By definition, α^* is an optimal projection given β^*. Similarly, β^* is an optimal solution to $\mathbf{ROUTING_{\Phi}}(\alpha^*)$. The KKT conditions for $\mathbf{ROUTING_{\Phi}}$ imply that the definition of optimal projection holds when $\alpha^*_{ic} > 0$. When $\alpha^*_{ic} = 0$, our choices of β^*_{jc} strictly follows the optimal projection definition. Combining the two cases shows β^* is an optimal projection given α^*. Therefore, (α^*, β^*) is a Nash equilibrium. ∎

B.3 Optimality of Nash Equilibria

Theorem 7. *Let (α^*, β^*) be a Nash equilibrium. Then $\{\alpha^*, \beta^*\}$ is an optimal solution to* $\mathbf{GLOBAL_{\Phi}}$.

Proof. Construct a set of variables $X^*_{ijc} = \alpha^*_{ic} \cdot \beta^*_{jc}$ for $\forall (i, j, c)$. We show that $\{X^*_{ijc}\}$ is an optimal solution to $\mathbf{GLOBAL}_X$ (when refined with link utilization penalties instead of hard link capacity constraints), given that $\{\alpha^*_{ic}, \beta^*_{jc}\}$ is a Nash equilibrium. It is easy to check that $\{X^*_{ijc}\}$ are feasible for (refined) $\mathbf{GLOBAL}_X$. Then, it suffices to show that the KKT conditions for $\mathbf{GLOBAL}_X$ hold with our choice of $\{X^*_{ijc}\}$ and appropriate choices of other parameter, *e.g.,* Lagrange multipliers, involved in the optimality conditions (which we discuss shortly). In a slight abuse of notation below, we write $f_{ijc}(X) = \mathrm{vol}_c(\mathrm{price}_j + K \cdot \mathrm{perf}_{jc}) / \sum_{c'} \mathrm{vol}_{c'} + \Phi'_j(\sum_c X_{ijc}\mathrm{vol}_c) \cdot \mathrm{vol}_c/|J|$. Here we consider the mapping constraint (3b) only and (3c) should follow similarly. We write down the *KKT conditions* for the refined $\mathbf{GLOBAL}_X$. There exist dual optimal variables $\{\lambda_{i1} \ge 0, \lambda_{i2} \ge 0\}_i$ such that

$$\begin{cases} f_{ijc} + (\lambda_{i1} - \lambda_{i2}) \cdot g_c \le f_{i'j'c} + (\lambda_{i'1} - \lambda_{i'2}) \cdot g_c, \text{ if } X_{ijc} > 0, \ \forall c \\ \lambda_{i1} \cdot \left(\sum_{jc} X_{ijc} g_c - w_i - \epsilon_i \right) = 0, \ \forall i \\ \lambda_{i2} \cdot \left(\sum_{jc} X_{ijc} g_c - w_i + \epsilon_i \right) = 0, \ \forall i \end{cases}$$

where $g_c = \mathrm{vol}_c / \sum_{c'} \mathrm{vol}_{c'}$. The first condition originates from the stationarity requirement, and can be interpreted as follows: when a link $j \in J_i$ (link j of data center i) is used to reach client c, *e.g.,* $X_{ijc} > 0$, its associated *marginal cost* must be no greater than the marginal cost of any other link $j \in J$. Note that the marginal cost conditions are similar to those in [7], but slightly more complicated due to the presence of dual variables and constraints. The second and third optimality conditions are complementary slackness for the dual variables and corresponding constraints.

Next we consider a Nash equilibrium (α^*, β^*). We can write down the KKT optimality conditions for $\mathbf{MAPPING_\Phi}$ and $\mathbf{ROUTING_\Phi}$, respectively, because the definition of optimal projection implies the optimality to the two optimization problems. Following the same notations, we have:

(i) *KKT optimality conditions* for $\mathbf{MAPPING_\Phi}$: there exist dual optimal variables $\{\mu_{i1} \geq 0, \mu_{i2} \geq 0\}_i$ such that

$$\begin{cases} \sum_{j \in J_i} \beta^*_{jc} f_{ijc} + (\mu_{i1} - \mu_{i2}) g_c \leq \\ \sum_{j' \in J_{i'}} \beta^*_{j'c} f_{i'j'c} + (\mu_{i'1} - \mu_{i'2}) g_c, \text{ if } \alpha^*_{ic} > 0, \ \forall c \\ \mu_{i1} \cdot \left(\sum_c \alpha^*_{ic} g_c - w_i - \epsilon_i \right) = 0, \ \forall i \\ \mu_{i2} \cdot \left(\sum_c \alpha^*_{ic} g_c - w_i + \epsilon_i \right) = 0, \ \forall i \end{cases} \tag{8}$$

Similarly, the first condition is the stationarity requirement, and the second and third conditions are the complementary slackness for the dual variables.

(ii) *KKT optimality conditions* for $\mathbf{ROUTING_\Phi}$:

$$f_{ijc} \leq f_{ij'c} \text{ for } j, j' \in J_i \text{ and } \beta^*_{jc} > 0, \ \forall (i,c) \text{ where } \alpha^*_{ic} > 0 \tag{9}$$

It follows that the values of f_{ijc} are equal for all j such that $\beta^*_{jc} > 0$ and $\alpha^*_{ic} > 0$.

By our construction, $X^*_{ijc} = \alpha^*_{ic} \cdot \beta^*_{jc}$. We next show that $\{X^*_{ijc}\}$ satisfy the optimality conditions (7), given the KKT conditions (8)-(9) established for $\{\alpha^*_{ic}, \beta^*_{jc}\}$. Without loss of generality, consider $X^*_{ijc} > 0$ for some (i,j,c) and $j \in J_i$, and we have $\alpha^*_{ic} > 0, \beta^*_{jc} > 0$.

From $\mathbf{ROUTING_\Phi}$ optimality conditions (9), we know that $f_{i\bar{j}c}$ takes the same value for all $\bar{j} \in J_i$ when $\beta^*_{\bar{j}c} > 0$. By our choice of $\beta^*_{jc} > 0$, we have

$$\sum_{\bar{j} \in J_i} \beta^*_{\bar{j}c} f_{i\bar{j}c} = f_{ijc} \cdot \sum_{\bar{j} \in J_i} \beta^*_{\bar{j}c} = f_{ijc} \tag{10}$$

Consider any link j' in data center i', *i.e.*, $j' \in J_{i'}$. There exists at least one link $j'' \in J_{i'}$ such that $\beta^*_{j''c} > 0$. Applying the optimality conditions (9) on (i', j'', c), we have

$$f_{i'j''c} \leq f_{i'j'c} \tag{11}$$

Further, by an argument similar to (10), we have

$$f_{i'j''c} = \sum_{\bar{j} \in J_{i'}} \beta^*_{\bar{j}c} f_{i'\bar{j}c} \tag{12}$$

To show that (7) holds for $\{X^*_{ijc}\}$, it suffices to find a set of parameters $\{\lambda_{i1} \geq 0, \lambda_{i2} \geq 0\}$ for all i such that those equalities and inequalities in (7) hold. We claim that the dual optimal variables $\{\mu_{i1}, \mu_{i2}\}$ in (8) can be used as a choice of $\{\lambda_{i1}, \lambda_{i2}\}$. We then show that the KKT optimality conditions (7) are satisfied with such choices. Consider $X^*_{ijc} > 0$ for some (i,j,c) and $j \in J_i$, we write down its associated marginal cost f_{ijc}, and compare it against that of any other link $j' \in J_{i'}$ for the same client c. We have

$$\begin{aligned} & f_{ijc} + (\lambda_{i1} - \lambda_{i2}) \cdot g_c \\ = \ & f_{ijc} + (\mu_{i1} - \mu_{i2}) \cdot g_c && \text{(choice of dual variables)} \\ = \ & \sum_{\bar{j} \in J_i} \beta^*_{jc} f_{i\bar{j}c} + (\mu_{i1} - \mu_{i2}) \cdot g_c && \text{by (10)} \\ \leq \ & \sum_{\bar{j} \in J_{i'}} \beta^*_{\bar{j}c} f_{i'\bar{j}c} + (\mu_{i'1} - \mu_{i'2}) \cdot g_c && \text{by (8)} \\ = \ & f_{i'j''c} + (\mu_{i'1} - \mu_{i'2}) \cdot g_c && \text{by (12)} \\ \leq \ & f_{i'j'c} + (\mu_{i'1} - \mu_{i'2}) \cdot g_c && \text{by (11)} \\ = \ & f_{i'j'c} + (\lambda_{i'1} - \lambda_{i'2}) \cdot g_c && \text{(choice of dual variables)} \end{aligned}$$

So we arrive at the conclusion that $\{X^*_{ijc}\}$ are optimal solutions to (refined) $\mathbf{GLOBAL}_X$. Therefore, $\{\alpha^*_{ic}, \beta^*_{jc}\}$ are also optimal solutions to $\mathbf{GLOBAL_\Phi}$ as they attain the same optimal objective values. ∎

B.4 Convergence to Nash Equilibrium

Theorem 8. *Alternate projection of* $\mathbf{MAPPING_\Phi}$ *and* $\mathbf{ROUTING_\Phi}$ *converges to a Nash equilibrium.*

Proof. Consider the function $\mathbf{obj} := \mathbf{cost} + K \cdot \mathbf{perf_\Phi}$, which is the common objective for both $\mathbf{MAPPING_\Phi}$ and $\mathbf{ROUTING_\Phi}$ problems. By the definition of optimal projections, the sequence of objective values of **obj** is *decreasing* under alternating optimizations of mapping and routing. In addition, the objective value is lower-bounded by the optimal value of $\mathbf{GLOBAL}_X$. Therefore, there exists a limit point of the sequence. Similarly, we can argue that the (α, β) sequence also has a limit point, since the feasible space of $\{\alpha, \beta\}$ is compact and continuous. It is not difficult to check that the limit point must be be a Nash equilibrium, as otherwise we can find another feasible solution that is within an ϵ-ball of the the limit point, which gives a lower objective value than the limit of **obj**, which contradicts our assumption at the beginning. ∎

C. PERFORMANCE PENALTY FUNCTION

ISP traffic engineering models link congestion cost with a convex increasing function $\Phi(r_j)$ on the link traffic load, *i.e.*, r_j. The exact shape of the function is not important, and we use the same piecewise linear cost

function as in [21], given below:

$$\Phi_j(r_j, cap_j) = \begin{cases} r_j & 0 \le r_j/cap_j < 1/3 \\ 3r_j - 2/3 \cdot cap_j & 1/3 \le r_j/cap_j < 2/3 \\ 10r_j - 16/3 \cdot cap_j & 2/3 \le r_j/cap_j < 9/10 \\ 70r_j - 178/3 \cdot cap_j & 9/10 \le r_j/cap_j < 1 \\ 500r_j - 1468/3 \cdot cap_j & 1 \le r_j/cap_j < 11/10 \\ 5000r_j - 16318/3 \cdot cap_j & 11/10 \le r_j/cap_j < \infty \end{cases}$$

In our evaluations, we scale this function such that 100% utilization gives a 250ms queuing latency, corresponding to some standard router buffer sizes [29].

D. EXPERIMENT SETUP

We evaluate the benefits of the joint mapping-routing approach using trace-based simulations for CoralCDN [18], a caching and content distribution platform running on top of Planetlab. Planetlab [30] is a service deployment platform which spans more than 800 sites geographically distributed in six continents. We correlate the traffic data with latency measurements from iPlane [15] which collects various wide-area network statistics to destinations on the Internet from a few hundred vantage points (hosted on Planetlab nodes). All data corresponds to March 31st, 2011.

Emulating data centers by clustering Coral nodes: Most Planetlab sites are single-homed *i.e.*, have just one ISP connecting them to the Internet. Therefore, it is challenging to demonstrate the benefits of our scheme on Coral, because the only decision that controls the latency and cost of user requests is the mapping decision. Instead, we adopt the approach followed in [2] where we treat multiple Planetlab sites in the same general metropolitan area as different egress links of a single data center located in that area. The density of Planetlab sites around certain cities makes such "clustering" of sites into data centers possible. We handpicked 12 data centers located in North America, Europe, Asia and South America, with a minimum of three Coral nodes in each data center and a maximum of six.

We show the locations of the data centers in Figure 11.

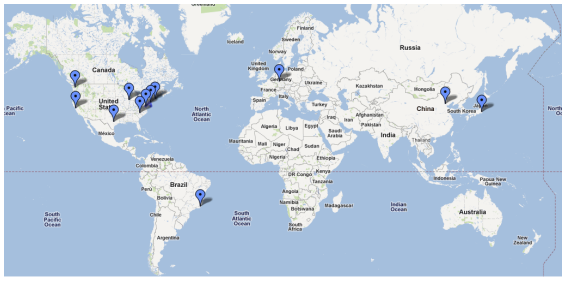


Figure 11: Location of the 12 emulated data centers.

Routable prefixes as client aggregates: We aggregate users into routable Internet prefixes (from a RouteViews [31] FIB dump), for two reasons. First, routing decisions at data center egress routers happen at this granularity, hence it is a natural choice. Second, the aggregation of IP prefixes in routing tables enables us to decide on reachability and latency information for a larger number of client IPs from the same set of initial latency measurements. The number of routable IP prefixes is large ($\approx 350,000$) but only ≈ 95000 of these send any traffic to Coral.

Traffic data from CoralCDN: We use a worldwide request trace from CoralCDN which lists a request timestamp, the Planetlab site at which it was received, a client IP address, and the number of bytes involved in the transfer. Overall, this trace contains 27 million requests and over a terabyte of data. For our experiments, we aggregate the requests made by clients into their most specific IP prefix from the RouteViews dump, and average their request rates for each hour.

Latency measurements from iPlane: We extract round trip latency (in milliseconds) from the iPlane logs, which contain traceroutes made to a large number of IP addresses from various vantage points at Planetlab sites. We only use latency information from vantage points which are also 'egress links' in the data centers we picked in Figure 11, which limits us to 49 vantage points. We assign the latency to each destination IP address to the corresponding most-specific IP prefix from the RouteViews dump, assuming that the latency is representative. If there is latency data for multiple IP addresses from the same prefix, we use their average value. The logs provide only one set of latency values for the entire day, and we assume that the paths are lightly loaded when the probes occur—hence treating these measured latencies as path propagation delays.

Picking a set of workable client prefixes: The set of destination IP prefixes for which latency data is available is not uniform across vantage points. Hence, we only retain data for destinations which are reachable from at least one vantage point belonging to each data center. This allows us to perform mapping of any client to any data center. Next, we intersect this set of destinations with those client prefixes which send traffic to Coral CDN at any time through the day. We are left with a set of about 24500 prefixes, which constitute about 39% of the total traffic trace by requests and 38% by bytes.

Capacity estimations: The capacities of links connecting Planetlab sites to the Internet are not publicly available, and even if they are, Planetlab machines are shared across multiple services each possibly with bandwidth caps. Instead of attempting to determine capacity or bandwidth caps from ground truth (this information is not publicly available to our knowledge), we

Data Center Popularity (total request volume per day: $\times 10^5$)	**Pricing** ($ per GB)
< 1	0.005
1 - 10	0.120
10 - 50	0.210
50 - 150	0.280
150 - 500	0.330

Table 4: Performance-based pricing model.

estimate them through traffic volumes in the trace.

The key assumption we make for setting capacities from observed traffic is that links are provisioned for a specific peak utilization. We determine the peak request rates received by Coral nodes which are part of the emulated data centers. We scale these numbers as follows: (1) we account for a specific peak utilization (80%), hence scaling by 100/80; (2) we account for the traffic received by *all* Coral nodes (including those not part of the emulated data centers) through scaling by the ratio of peak hourly traffic over all Coral nodes to the peak hourly traffic over the chosen Coral nodes.

Bandwidth cost estimation: We assume a simple charging model based on total amount of data transferred over a given period of time. We employ a *performance-based* model where we assign a higher cost to a link which provides lower latency paths to a higher proportion of traffic volume (as determined from the traffic and latency trace information). To quantify this, we first determine the number of requests serviced by each Coral node assuming that clients are serviced by the latency-wise closest Coral node. We then use table 4 (motivated by price values from Amazon EC2 bandwidth pricing [22]) to determine the charging price per GB of data sent over that link.