Redesigning CDN-Broker Interactions for Improved Content Delivery

Matthew K. Mukerjee[⋆], Ilker Nadi Bozkurt[†], Devdeep Ray[⋆],
Bruce M. Maggs[†‡], Srinivasan Seshan[⋆], Hui Zhang[⋆§]
[⋆]Carnegie Mellon University, [†]Duke University, [‡]Akamai Technologies, [§]Conviva Inc.

ABSTRACT

Various trends are reshaping Internet video delivery: exponential growth in video traffic, rising expectations of high video quality of experience (QoE), and the proliferation of varied content delivery network (CDN) deployments (e.g., cloud computing-based, content provider-owned datacenters, and ISP-owned CDNs). More fundamentally though, content providers are shifting delivery from a single CDN to multiple CDNs, through the use of a content broker. Brokers have been shown to invalidate many traditional delivery assumptions (e.g., shifting traffic invalidates short- and long-term traffic prediction) by not communicating their decisions with CDNs. In this work, we analyze these problems using data from a CDN and a broker. We examine the design space of potential solutions, finding that a marketplace design (inspired by advertising exchanges) potentially provides interesting tradeoffs. A marketplace allows all CDNs to profit on video delivery through fine-grained pricing and optimization, where CDNs learn risk-adverse bidding strategies to aid in traffic prediction. We implement a marketplace-based system (which we dub Video Delivery eXchange or `VDX`) in CDN and broker data-driven simulation, finding significant improvements in cost and data-path distance.

CCS CONCEPTS

• **Networks → Control path algorithms**; *Network protocol design*; Application layer protocols;

KEYWORDS

CDNs; content brokers; content delivery; interfaces

ACM Reference Format:
Matthew K. Mukerjee, Ilker Nadi Bozkurt, Devdeep Ray, Bruce M. Maggs, Srinivasan Seshan, Hui Zhang. 2017. Redesigning CDN-Broker Interactions for Improved Content Delivery. In *Proceedings of CoNEXT '17*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3143361.3143366

1 INTRODUCTION

Content delivery is constantly changing to meet the evolving challenges created by new workloads (e.g., streaming video), new actors (e.g., CDNs), new protocols (e.g., HTTP chunk-based video), new algorithms (e.g., video rate adaptation), and new demand (e.g., exponential growth in video traffic). Techniques introduced to accommodate these challenges have far-reaching repercussions on flows across all layers of the network stack. For example, the introduction of content delivery networks (CDNs) dramatically changed the traffic patterns that ISPs handled, clients' performance expectations, and the sheer volume of content that the Internet could deliver.

Content delivery is in the midst of another such major change. Until recently, major content providers (CPs) either contracted with a single CDN, such as Akamai [51], Level 3 [3], or CloudFront [9], or deployed their own CDN, such as Google [25] and Netflix [46]. The recent rise of CDN management services ("brokers"), such as Cedexis [18], Conviva [2], or NicePeopleAtWork [50], and CDN federation techniques [1, 49] has made it easier for content providers to enlist multiple CDNs to deliver content. Simultaneously, the rise of ISP CDNs (e.g., Comcast [11]) and proposals like virtualized CDN nodes running inside of ISPs [21, 22], are moving previous ISP-CDN tussle concerns [22, 34, 35, 54, 55] into the new context of a CDN-broker tussle.

At first glance, it may seem that the addition of brokers to content delivery is a minor change; however, brokering is a surprisingly complicated process. Our previous work [45] uses CDN and broker data to show there are significant issues in today's content brokering ecosystem, due to the lack of CDN-broker coordination in optimizing delivery objectives (e.g., cost, performance, etc.), that require fundamental changes. These issues, however, have not been widely identified as they are hard to diagnose without both broker and CDN data. They have also yet to become widespread, as broker traffic is still a small (but growing) portion of overall CDN traffic.

In this paper, we first characterize the kinds of problems brokers and CDNs face due to independent decision making by examining data from both a popular broker and a major CDN. Among the problems we uncover is that brokers may make traffic unpredictable for CDNs, making it difficult for CDNs to profit, due to combination of long-term CDN-CP contracts (months or years [7]) and flat-rate pricing. Additionally, despite having multiple clusters with similar performance, CDNs have no incentives to share this information with brokers today, limiting a broker's ability to optimize for certain CP goals, and to handle failures.

These problems lead us to three requirements needed for proper CDN-broker decision making: 1) CDNs need to replace today's flat-rate price model and reflect dynamic per-cluster prices to improve profitability; 2) CDNs need incentives for providing a fine-grained cluster-level view to brokers, allowing them to better optimize for CP goals; and 3) CDNs and brokers need to make decisions jointly, removing today's traffic unpredictability, and improving stability in the content delivery ecosystem.

CoNEXT '17, December 12–15, 2017, Incheon, Korea

ACM ISBN 978-1-4503-5422-6/17/12…$15.00
https://doi.org/10.1145/3143361.3143366

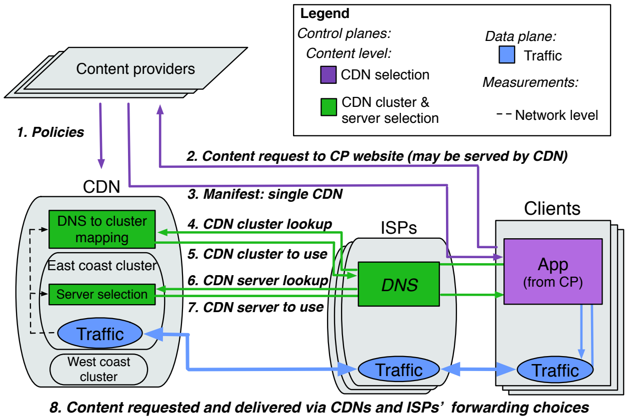

Figure 1: Traditional content delivery.

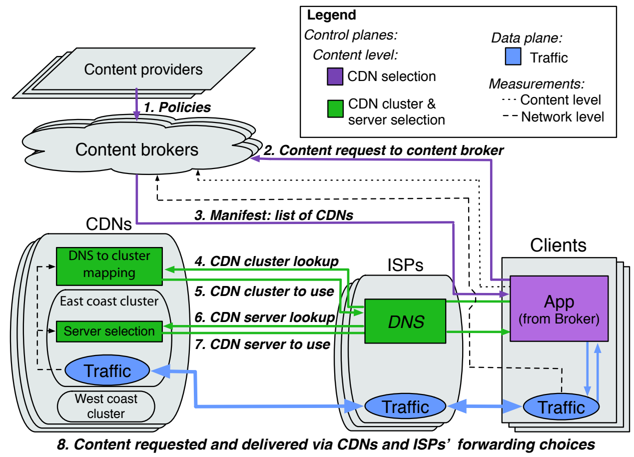

Figure 2: Brokered content delivery.

Solutions that only address one of these requirements do not provide the right adoption incentives for CDNs, brokers, and CPs; CDNs only benefit from dynamic cluster-level pricing and traffic stability, but Brokers/CPs only benefit from cluster-level optimization. Addressing all the requirements simultaneously provides incentives to all parties. While similar to the well studied ISP-CDN collaboration problem [22, 34, 35, 54, 55], we argue CDN-broker collaboration is easier to achieve, as there are significantly fewer CDNs than ISPs, and business relationships are already more attuned to collaboration (CDNs and brokers both directly optimize content delivery under contract with CPs).

We address the above requirements directly by examining promising points in the design space. Simple tweaks to today's practices (e.g., providing brokers multiple clusters to choose from) do not meet all the requirements (and thus lack deployment incentives). In addition, multiparty transaction designs requiring all CDNs and brokers to agree are impractical. We find that a marketplace-like design represents a reasonable tradeoff. It meets the first two requirements while allowing CDNs to learn "bidding" strategies that likely provide them traffic predictability. A marketplace represents one possible solution, however, the focus of this work is that all parties (including clients) benefit from a content delivery service sold at much finer granularity than today. Any mechanism that supports these requirements may be sufficient.

We present a prototype marketplace design called Video Delivery eXchange (VDX). We leverage real-world traces obtained from a major CDN and a popular broker, as well as publicly available data from other CDNs, to build a CDN-scale simulation. We run our simulator across a variety of scenarios (e.g., differences in CDN deployment models, differences in country pricing) to better understand how relatively complex schemes like VDX can fine-tune the trade-off between performance and cost.

To summarize, we make the following contributions:

1. Identify the challenges created by the lack of joint decision making between brokers and CDNs by analyzing broker and CDN data.
2. Examine the design space, from simple tweaks (e.g., dynamic pricing or providing multiple potential clusters to brokers) to more complex designs (e.g., marketplace or multiparty transactions), evaluating their tradeoffs.
3. Evaluate a marketplace design (VDX), where all CDNs can profit on video delivery, in depth through CDN-scale simulations using data from both a broker and a CDN, finding significant improvements in cost and data-path distance.

2 CONTENT DELIVERY: THE PAST AND THE PRESENT

The arrival of CDNs has had a dramatic impact on the Internet. In this section, we explain how content delivery is again being reshaped for content providers by contrasting broker-based delivery with traditional CDN delivery.

2.1 Traditional Content Delivery

Content providers (CPs), such as ESPN, Netflix, and HBO, create or license content that users are interested in. In order to provide good "quality of experience" (QoE) (e.g., a combination of metrics such as average bitrate, buffering ratio, and join time [13]) to viewers around the globe, CPs would need to build massive amounts of infrastructure. Thus, most CPs rely on CDNs to provide reasonable QoE. CPs generate revenue through premium services and/or advertising, and try to minimize their delivery costs. CPs often pay CDNs based on bandwidth usage based on a 95/5 model [43].

CDNs deliver content to clients through clusters nearby (e.g., in datacenters, peering points, universities, large businesses, or ISP networks) to minimize latency and improve throughput. CDNs have a wide variety of deployment models: some deploy servers in a large number of geographic regions (e.g., Akamai [51]); others deploy in a small set of strategic regions (e.g., Level 3 and CloudFront) [37]; other "ISP CDNs" operate extremely locally, serving a single ISP's customers in a region (e.g., Comcast [11]). CDNs typically choose which cluster to serve a client request from based on network measurements or static assignments. Akamai, for example, uses latency and loss measurements from clusters to gateway routers in the network (not individual clients) [19, 38] to decide on an initial cluster assignment.

CDNs wish to provide reasonable performance to clients while minimizing their bandwidth and co-location (energy) costs. In a recent annual report [4], Akamai lists bandwidth costs as their largest cost ($150M/year) behind payroll, with slightly lower co-location costs ($126M/year). CDNs generally do not price their services to reflect costs at individual server locations (which may vary considerably; see §3.1), and, instead, use a flat-rate price across large geographic regions (e.g., continents) [10, 44] regardless of the actual delivery cost. Prices vary as CDNs typically negotiate individual contracts with CPs over long timescales (e.g., months, years [7]). This lack of fine-grained cost-aware pricing can lead to significant problems, as we show in §3.2.

Traditionally (see Figure 1), CPs contract with a single CDN and express very broad policy goals (e.g., what content can be served by the CDN). The CDN typically caches the content on front-end servers close to the clients (although more complicated caching structures also exist). Clients request an HTML page or video manifest from the CP's website that indicates which CDN to contact for the content. The CDN chooses which server to use for a given client, and provides a mechanism (e.g., DNS) for reaching the server. The client connects to this server and retrieves the content.

2.2 Brokers and Delivery Today

With increasing pressure from users in terms of QoE expectations, as well as the sheer volume of traffic, CPs have moved from a single-CDN delivery model to employing multiple CDNs [14, 56]. Due to variations in price and performance, spatially and temporally [39], CDN selection must be dynamic. Figure 2 shows brokered content delivery. From a client's perspective, brokers are a level of indirection for CDN selection; clients first ask a CP's broker which CDN to use, before querying a CDN's DNS server.

Brokers (e.g., Cedexis [18], Conviva [2], NPAW [50]) measure QoE within client applications (e.g., video players) and build predictive models to determine the best assignments of clients to CDNs (using CPs' QoE and cost goals) based on various factors (e.g., client's location, ISP, etc.) [23, 33]. Brokers not only select the initial CDN a client is assigned to, but also move clients between CDNs in real-time (e.g., mid-stream). Although a CP could function as a broker for its own content, independent brokers can leverage data across CPs' clients and CDNs to form a more complete view.

There is no explicit coordination between brokers and CDNs, a key point of tension addressed in this paper (see §3). CDNs, however, implicitly see the effects of brokers' decisions when clients are suddenly moved to/away from their clusters. This increased traffic unpredictability, along with long-term CDN-CP contracts still based on flat-rate pricing, potentially makes the disparity between prices and internal CDN costs even worse. (see §3.2). Brokering also facilitates a wider variety of small-scale deployment models (e.g., regional CDNs, city-centric CDNs, etc.), although we have yet to see these types of CDNs in practice.

3 POTENTIAL PROBLEMS AND OPPORTUNITIES

Although brokers may appear to add a simple layer of indirection, they greatly complicate content delivery. Brokers and CDNs run independent control loops to maximize their own objectives, without explicitly communicating their decisions. These decisions directly impact one another, potentially leading to sub-optimal decisions for both parties.

Brokers have a global view of all client performance (app-level QoE) and costs for a CP, but can only make decisions by selecting which CDN to send a client to. Individual CDNs, conversely, have a large set of clusters to choose from but typically make their choice only on network level measurements, rather than QoE (each CDN would need to instrument CPs' software to get QoE). This mismatch of data richness (brokers) and selection richness (CDNs) leads to many potential problems.

We examine these problems separately, using data from a major CDN and a popular broker. The broker data allows us to understand when it uses different CDNs (e.g., over geographic regions, time, etc.). The CDN data allows us to understand its use of different server clusters. We distill these problems down to a short list of key requirements any proposed CDN-broker decision interface must meet to aid in CDN pricing, meet flexible performance goals, and provide traffic stability.

3.1 Traces

Broker: We collect trace data from a video delivery broker. The trace includes an entry for each client session containing the request arrival time, which video was requested, the average bitrate, session duration, the client city and AS, the initial CDN contacted, and the current CDN delivering the video. The data covers roughly an hour of off-peak requests (33.4K total) for one content provider (a music video streaming website). Even this small window illustrates many problems.

The data exhibits similar trends to those seen in other works [12]: video popularity follows a Zipf distribution, and the distribution of client cities follows a power-law. Most clients abandon almost immediately (around 78%). The distribution of bitrates is bimodal with peaks at the lowest and highest bitrate. The trace identifies three large CDNs (here "A," "B," and "C") directly and lists the rest as "other." CDN A is a CDN with clusters in many locations. CDN B and C deploy large amounts of capacity in a small number of locations. We investigate the effects of different deployment models in our evaluation (§7).

CDN: We collect Internet mapping data from a major CDN to compare performance estimates across its clusters. The data provides a score estimating the performance between blocks of client IP addresses and candidate CDN clusters. This score is a simple function of latency and packet loss. Measurement happens periodically and frequently (several times per minute) through pings from clusters to routers with large networks of clients behind them.

From the same CDN, we collect data on the average cost per byte delivered for the 20 countries with the highest volume of traffic, using client geolocation to bin requests into countries. We then compare them to the average delivery cost. We anonymize this data and present it in Figure 3.

3.2 Potential Problems for a CDN

Brokers create problems for CDNs for three main reasons: 1) brokers make load balancing difficult due to short-term traffic unpredictability, 2) brokers make provisioning difficult due to long-term traffic

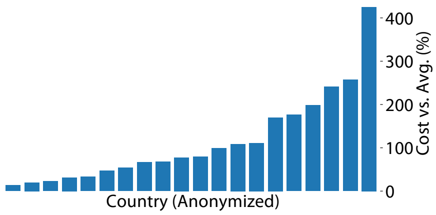

Figure 3: Average cost per byte serving clients geolocated in various countries relative to the average.

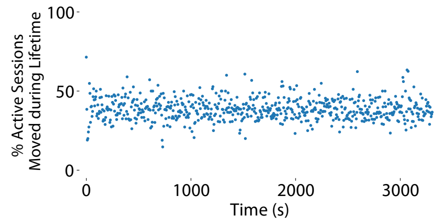

Figure 4: Sessions moved between CDNs by the broker in our trace in 5s intervals.

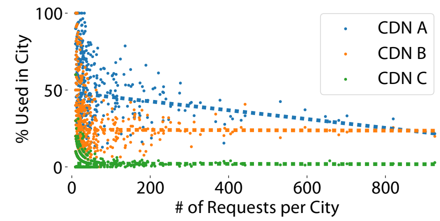

Figure 5: Broker's usage of CDNs, sorted by requests per city in the US. Dotted lines are best-fit linear regressions.

unpredictability, and 3) (broker-created) traffic unpredictability negatively impacts profits due to flat-rate CDN-CP contracts.

Short-term provisioning problems due to traffic unpredictability: Figure 4 shows a time-series graph of the percentage of client sessions in the broker trace, within 5 second intervals, that have been shifted from one CDN to another mid-stream. This value is surprisingly high throughout (averaging ~40%). We note, however, that at some points this dips to ~20% and at other times rises above ~60%. This indicates brokers not only often move traffic around, but the rate at which they do is highly variable. This potentially makes short-term provisioning (load balancing) difficult for CDNs.

Long-term provisioning problems due to traffic unpredictability: Figure 5 shows the utilization for the CDNs in the broker trace, plotted as a function of number of requests per city. The dotted lines are best-fit lines. We see from the best-fit lines that regardless of city size, CDN B and CDN C's usage does not change, whereas CDN A is strongly favored in smaller cities. This is perhaps unsurprising due to CDN A's broader geographic deployment. CDN A is also generally more expensive than CDN B and CDN C, suggesting that a broker will try to avoid CDN A where other options are available. Succinctly, brokers do not merely split traffic evenly among CDNs; traffic may be arbitrarily divided in geographic regions due to various factors and change over time.

This leads us to believe that brokers, as well as other CDNs, can cause a CDN difficulty in cluster planning and long-term provisioning (e.g., cluster location and capacity). For example, if a broker decides to stop using CDN A in big cities (e.g., CDN B deploys more servers) this will impact CDN A's future provisioning. If CDN B then raises its prices, the broker may move more traffic back to CDN A, again impacting future provisioning. In effect, in a brokered world, proper CDN provisioning becomes more difficult to achieve.

> **REQUIREMENT: Traffic Predictability**
> In order to provide more stability for CDNs, broker-controlled traffic must be more predictable. Thus, a proper CDN-broker decision interface must make decisions jointly (i.e., share information and decisions in both directions).

Pricing / cost disparities: Figure 6 illustrates a toy example of CDN pricing issues. Recall that CPs generally pay CDNs a contracted flat rate per traffic delivered (based on a 95/5 model [16, 43]) with price changes (e.g., 2 – 7×) depending on very coarse geographic regions (e.g., continents) [10, 44]. CDN Y can provide good performance at a low flat-rate contract price for the CP, for all

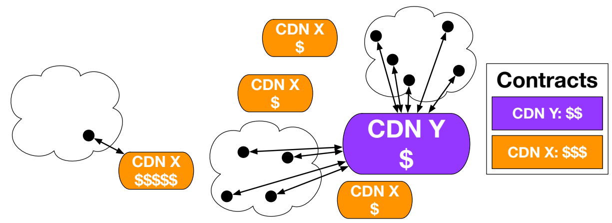

Figure 6: Brokers can greatly skew CDN traffic patterns making it difficult for CDNs to profit. Ovals represent CDN clusters, with "$" indicating the cost to the CDN. Squares represent CDN-CP contracts, with "$" indicating the price paid by the CP. Solid dots represent clients.

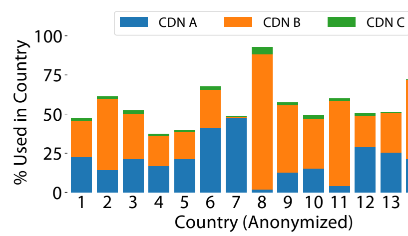

Figure 7: Broker's usage of CDNs for a sampling of countries based on request count.

clients except for the left-most one, who must be served by CDN X. Unfortunately for CDN X, this client is served by a very expensive cluster. CDN X's flat-rate contract price with the CP is more expensive than CDN Y, so a broker (unknowingly) avoids CDN X's cheaper clusters. This unfortunately means that CDN X actually loses money as the CP will pay CDN X at a price less than its cost.

We see potential pricing problems like this occur at the country level. Figure 7 shows how the utilization of the CDNs in our broker trace differ in different countries. The remaining percentage of clients are serviced by other smaller CDNs. We show all countries that originated 100 or more requests in our trace, in random order. Note that utilization varies significantly: e.g., CDN B barely serves 7, yet almost entirely serves 8; CDN A is rarely used in 8, 11, and 15, etc.

Different countries around the globe can have markedly different bandwidth costs. Our CDN cost data shows up to a ~30× disparity in pricing between countries (Figure 3). CDN CloudFlare paints a similar picture, stating that when compared to Europe, North America, Asia, Latin America, and Australia, cost 1.5×, 7×, 17×, and 21× more respectively [20]. They further state that within a region, some transit ISPs may have an order of magnitude higher cost.

1 Alternative Choice	2 Alts.	3 Alts.	4 Alts.
77.8%	64.5%	53.7%	43.8%

Table 1: How often alternative CDN clusters with similar performance scores exist.

If, for example, Country 7 in Figure 7 is very expensive for a CDN, yet Country 8 is very cheap (with flat-rate pricing across both), CDN A will have trouble making a profit, whereas CDN B will easily make a profit. Therefore, unpredictable traffic from brokers may unintentionally cause disparities between pricing and cost, affecting CDN profits. Unfortunately, raising contract prices to recoup these profits will likely cause brokers to move even more traffic away from the CDN.

REQUIREMENT: Dynamic Cluster Pricing
In order to alleviate CDN pricing / cost disparities, a proper CDN-broker decision interface must allow CDNs to charge CPs (or brokers) prices reflecting their internal costs. In order to allow CPs/brokers to optimize over these prices, a price sharing mechanism that is fine-grained both spatially (per-cluster, not per-continent) and temporally (per-minute, not per-year) is needed.

3.3 Potential Problems for a Broker

CDNs create problems for brokers for two main reasons: 1) CDNs do not expose cluster-level information to brokers, limiting a brokers' ability to optimize performance and cost, and 2) CDNs make decisions that impact end-to-end delivery without information about all clients seen by brokers, directly affecting performance.

Limited optimization ability from lack of cluster-level view: CDNs and brokers do not explicitly consult with each other for decision-making; they lack an interface to facilitate this exchange. Brokers cannot gather much information about CDNs at a cluster level, and thus, treat each CDN as a black box function of {client location, client ISP, ...} → {performance, cost}. As CDNs typically map individual clients to one specific CDN cluster at a given time, when performance is inadequate based on the CP's objectives, a broker's only recourse is to switch CDNs (even if other better choices exist within the current CDN). Effectively, the granularity of change a broker can make is very coarse.

Table 1 shows how often there are alternative clusters with similar estimated performance (based on latency and loss measurements) in the CDN data. We find that on average there are four server clusters (i.e., 3 alternative choices) that have similar scores (within 25% of the best), yet typically only one choice is returned. This data indicates potential opportunities; as these clusters have similar performance estimates, brokers may be able to avoid switching CDNs due to inaccurate estimates, congestion, or failures (unlike today), to better meet CPs' performance goals. As explained above, today's flat-rate pricing discourages CDNs from making use of these alternative clusters if their costs are higher than the primary cluster.

Poor performance due to incomplete data: Both brokers and CDNs spend significant effort building maps of the Internet to predict performance between clients and servers. This is by no means a small task; in recent work [23] a broker claimed that they regularly handle 100M client sessions per day, 3M clients concurrently during peak hours, and 10s–100s of thousands of clients entering and exiting per minute. They also imply that this leads to 50–100 GB of new sample data to process per minute. Sharing mapping information could greatly improve the accuracy of the data as both CDNs and brokers have limited vantage points into the network. Namely, CDNs such as Akamai typically measure (in advance of connections) from clusters to gateway routers [38], whereas brokers generally only measure (during a connection) from clients to chosen CDN servers.

REQUIREMENT: Cluster-level Optimization
Brokers currently make decisions with an unnecessarily coarse view of CDNs, responding to issues by switching CDNs entirely, despite other reasonable clusters being available within the CDN. Clusters may also have different performance/cost tradeoffs leading to better CP goal optimization. Thus, a proper CDN-broker decision interface should provide brokers with cluster-level views of CDNs.

4 EXPLORING THE DESIGN SPACE

As seen, the problems from §3 motivate an explicit broker-CDN decision interface that provides: 1) cluster-level optimization, 2) dynamic cluster pricing, and 3) traffic predictability. We find many possible designs that meet some/all of these requirements have the same basic structure. While many of these designs are reasonable, we argue that only designs that meet all three requirements have adoption incentives for CDNs, brokers, and CPs.

4.1 Generalizing Designs

Building off today's CDN-broker interactions (as seen in §2), all designs we consider utilize the following two protocol structure:

Decision Protocol: Periodically (e.g., every few minutes, as brokers do today [23]), each design runs the seven steps below to update the mapping of clients to CDN (clusters) that maximizes CP goals (see Figure 8):

1. **Estimate**: CDN clusters estimate capacity, cluster costs, and cluster-to-client performance.
2. **Gather**: Brokers count clients, including meta-data (e.g., location).
3. **Share**: Brokers potentially send client (meta-)data to CDNs. We will see that designs that do not send client data make it difficult for CDNs to provide proper matchings (as they do not know which clients belong to which broker (or to no broker at all).
4. **Matching**: CDNs match clients to one or more potential clusters, based on performance estimates, cluster costs, and capacities, as well as any client (meta-)data from brokers. Current algorithms such as Akamai's customized stable marriage algorithm [40], or simpler algorithms, such as ranking clusters that can provide adequate performance by their costs, provide plausible starting points.

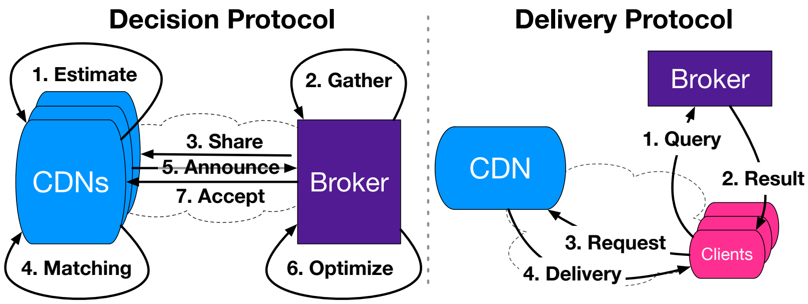

Figure 8: All CDN-broker joint-decision interfaces follow this basic structure, differing in how they implement steps.

5. **Announce**: Brokers receive CDNs' matching data (either pushed or pulled), and potentially performance, cost, and capacity information. We will see that designs that do not announce all three provide worse overall performance and cost.
6. **Optimize**: Brokers map clients to CDN (clusters) to meet CP goals, using the CDNs' matchings and current cost, capacity, and performance (including traditional application-level QoE) data/estimate. We show an example optimization ILP (to be solved by the broker), in Figure 9, that maximizes performance while minimizing cost (with weights w_p and w_c). The output is stored for the Delivery Protocol. Other optimizations are equally valid (e.g., utilizing client meta-data, or simpler greedy algorithms).
7. **Accept**: Brokers tells all CDNs which matchings were used, so CDNs can modify future matchings.

High-level designs differ solely on how they implement *Share*, *Matching*, and *Announce*. As brokers operate on behalf of CPs and can control client requests directly, all considered designs have the broker make the final decision.

Delivery Protocol: Whenever a client initiates content retrieval, the following protocol runs:

1. **Query:** Client queries broker for CDN (cluster).
2. **Result:** Broker returns CDN (cluster) chosen earlier.
3. **Request:** Client requests content from CDN (cluster).
4. **Delivery:** CDN (cluster) delivers data to the client.

Note that the most recent Decision Protocol results are used, and thus decision making does not slow down delivery. All designs use the same Delivery Protocol, thus we end our discussion of it here.

4.2 Design Space

We now present alternate designs for CDN-broker decision making interfaces that differ in *Share*, *Matching*, and *Announce*, and the requirements from §3 (see Table 2).

`Brokered` (today's world §2): . CDNs and brokers share little information, and CDNs match clients to single clusters. As we have seen in §3 this does not meet our *Cluster-level Optimization*, *Dynamic Cluster Pricing*, or *Traffic Predictability* requirements.

`Multicluster`: CDNs provide multiple similar cluster options per client. From this, brokers learn rough performance values. This provides *Cluster-level Optimization*, but does not address *Dynamic Cluster Pricing* concerns, or provide *Traffic Predictability*.

$$
\begin{aligned}
\max\ & w_p * \textstyle\sum_{r\in\text{Clients},\, m\in\text{Matchings}_r} \text{Performance}(m) * \text{U}_{r,m} \\
-\ & w_c * \textstyle\sum_{r\in\text{Clients},\, m\in\text{Matchings}_r} \text{Cost}(m) * \text{Bitrate}(r) * \text{U}_{r,m} \\
\textbf{subjec}&\textbf{t to:} \\
&\forall r \in \text{Clients}, m \in \text{Matchings}_r : \text{U}_{r,m} \in \{0,1\} \\
&\forall r \in \text{Clients} : \textstyle\sum_{m\in\text{Matchings}_r} \text{U}_{r,m} = 1 \\
&\forall l \in \text{Clusters} : \textstyle\sum_{r\in\text{Clients},\, m\in\text{Matchings}_{r,l}} \text{Bitrate}(r) * \text{U}_{r,m} \\
&\qquad \le \text{Capacity}(l)
\end{aligned}
$$

Figure 9: Example broker optimization problem.

`DynamicPricing`: CDNs share dynamic cost information with brokers. This fixes *Dynamic Cluster Pricing* concerns, but does not address *Cluster-level Optimization* or provide *Traffic Predictability*.

`DynamicMulticluster`: Combines `Multicluster` and `DynamicPricing`. This addresses both *Cluster-level Optimization* and *Dynamic Cluster Pricing*, but does not provide *Traffic Predictability*. Its major flaw is instability; as decisions are not made jointly, the clusters with the best performance-to-cost ratio are overwhelmed as specific cluster capacity values are unknown. This is similar to the instability seen in price-based routing schemes [24, 29, 36].

`BestLookup`: This design attempts to fix `DynamicMulticluster` by providing cluster capacity information to brokers. CDNs must build multiple potential client-to-cluster matchings without knowing which clients are being considered by the broker. If there are multiple brokers or significant non-broker traffic (as there is today), "overbooking" of traffic sources may still overwhelm capacity (e.g., a cluster with capacity 10 units may receive 9 units of traffic each from two brokers).

`Marketplace`: A marketplace-based design would view CDNs' matchings as *bids* for the brokers' resource (clients). When the Decision Protocol is run periodically, there is a single round of bidding for clients, in which all CDNs are first told about all clients (meta-)data. CDNs build more nuanced matchings, properly allocating capacity based on the received client data. Brokers optimize as before and return a list of accepted bids to CDNs. CDNs learn which bids are likely to be used over time (as they know which clients are associated with which broker and get explicit feedback on why bids fail), providing "weak" *Traffic Predictability*.

	Share	*Matching*	*Announce*	CO	DCP	TP	Runtime
`Brokered`		Single-Cluster					1 Round
`Multicluster`		Multi-Cluster	Performance	✓			1 Round
`DynamicPricing`		Single-Cluster	Cost		✓		1 Round
`DynamicMulticluster`		Multi-Cluster	Cost, Performance	✓	✓		1 Round
`BestLookup`		Multi-Cluster	Cost, Performance, Capacities	✓	✓		1 Round
`Marketplace`	Clients	Multi-Cluster	Cost, Performance, Capacities	✓	✓	Weak	1 Round
`Transactions`	Clients	Multi-Cluster	Cost, Performance, Capacities	✓	✓	Strong	Multi-Round

Table 2: Alternate designs for a CDN-broker decision making interface, and whether they meet the *Cluster-level Optimization* (CO), *Dynamic Cluster Pricing* (DCP), and *Traffic Predictability* (TP) requirements in §3.

`Transactions`: After *Optimize*, the broker requests CDNs to commit the resources for the chosen client-to-cluster mapping. If any CDN disapproves the mapping, the mapping is withdrawn from all CDNs and a new mapping is computed. This provides stronger *Traffic Predictability* guarantees than `Marketplace` by making the process transaction-like, however, it is unrealistic, as CDNs may never all approve the mapping. Thus, we do not consider it further.

5 NARROWING THE DESIGN SPACE

We present an evaluation of the different designs presented in §4. As it is not practical to deploy a multi-CDN-broker marketplace for evaluation, we focused on building a realistic simulator using CDN and broker data (§3.1), as well as other publicly available CDN data.

5.1 Simulation Overview

We simulate 14 world-wide CDNs and a broker focused on video delivery. We run one round of the Decision Protocol (§4.1) to determine our results, effectively building a "snapshot" of client-CDN cluster assignments. Time dynamics are less important as the Decision Protocol runs periodically (e.g., every few minutes) over all clients.

Clients: We use the client requests (with location and bitrate) from the broker data we received (§3.1). Client locations in the broker data are matched with client locations in the CDN data to allow us to use client-to-cluster performance (latency/loss) scores in the CDN data. Some client-clusters pairings do not have scores, so we extrapolate them by computing a linear regression of scores with respect to client-cluster distance.

We simulate an additional 3× this amount of clients as background traffic (e.g., other broker traffic or non-broker traffic) not optimized by this broker. While difficult to quantify (for both the CDN and broker), traffic today is predominantly non-brokered, but has been progressively changing.

Broker: We simulate a broker using the ILP in §4.2 as the optimization function, solved by Gurobi [28].

CDNs: Each CDN is defined by a list of cluster locations. We received world-wide cluster location information from one highly distributed CDN. We additionally inferred the locations of as many CDNs as we could find (13) on PeeringDB [53]. PeeringDB may underestimate cluster locations, but for the smaller CDNs we manually verify their locations based on information available on their websites.

CDN cluster locations and cost: Each cluster location has an associated bandwidth and co-location (energy) cost, expressed in dollars per bit. We generate bandwidth costs by choosing average costs for countries from the data in Figure 3, then assign bandwidth costs to specific clusters by drawing from a normal distribution centered on this mean, with standard deviation derived from CDN bandwidth cost data for the top 8 ISPs within the US. Co-location costs are based on the cost for the country, but decrease proportional to the logarithm of the number of CDNs in that location. This models the fact that more CDNs are located in places that are inexpensive to serve from.

CDN contract price and capacity: Each CDN has a contract price that we use in flat-rate price designs. A CDN's contract price is the average price per bit for the CDN if it was individually offered to all clients. Cluster capacity is assigned similarly; all clients are sent to each CDN individually and clusters are assigned 2× received traffic as their capacity. We assume that in steady-state, clusters are provisioned with ample capacity. Clusters that did not see any clients take capacity from their closest neighbor with capacity. Designs that do not share cluster capacity information with brokers use the median cluster capacity (per-CDN) as an estimate.

CDN matching algorithm and bidding: For each client, a CDN selects a set of candidate clusters with scores at most 2× worse than the best score. If there is no other cluster with a score within 2× the best, the second best scoring cluster is selected. Candidate clusters are sorted from lowest to highest cost, with the matchings prioritized in that order. The scores, costs, and capacities of CDNs are directly reflected in *Announce* (depending on the design) for simplicity. Real-world CDN matching algorithms could change over time to find risk-averse strategies. We avoid this for simplicity.

Designs: We compare the designs presented in §4.2. The *Matching* algorithm in `Multicluster` (2), (100), and `Marketplace`, produces 2, 100, and 100 alternative clusters respectively. `Omniscient` exposes all CDN data to the broker.

Metrics: We compare designs using *Cost*, *Score*, *Distance*, *Load*, and *Congested* as metrics. *Cost*, *Score*, and *Distance* are the median cost, score, and distance over all clients (lower is better). Load is the median cluster load over all CDN clusters that saw any traffic. Congested is the percentage of clients sent to clusters that have greater than 100% load.

	Cost	Score	Distance	Load	Congested
Brokered	136	132	297	9%	0%
Multicluster (2)	155	87	194	14%	27%
Multicluster (100)	171	85	141	20%	39%
DynamicPricing	126	148	318	11%	0%
DynamicMulticluster	115	122	219	40%	14%
BestLookup	94	108	166	14%	14%
Marketplace	93	112	178	23%	0%
Omniscient	86	111	172	48%	0%

Table 3: Comparing the different designs for various metrics in data-driven simulation. Lower values are better.

5.2 Results

In Table 3, we summarize the results. Brokered serves as our baseline; Both Multicluster designs provide better performance at the expense of increased cost (as the first cluster bid is always the cheapest one according to the CDN Matching algorithm above); additional clusters may provide better performance but will not be cheaper than the first cluster. Both designs also overload clusters (as these designs only estimate cluster capacities) while optimizing for performance. DynamicPricing marginally saves cost by dropping performance (without overloading clusters), by only exposing one cluster, but allowing the broker to optimize with knowledge of the CDN costs. This shows that just avoiding expensive clusters is not good enough; a balance of expensive and cheap clusters are needed provide good performance. DynamicMulticluster and BestLookup do better than Brokered in both performance and cost (with BestLookup spreading load better), but still overloads some clusters (given their inaccurate capacity info). Marketplace does very similar to BestLookup but avoids overloading any clusters as it has accurate capacity info. Omniscient provides similar results, with the lowest cost overall. We see the same trends in the CDFs of cost, score, and distance (not presented).

From these results, we see that BestLookup and Marketplace are promising points in the design space. However, Marketplace better meets the requirements from §3 as it is less likely to overload clusters. By addressing all the requirements, we argue that Marketplace is more aligned with adoption incentives than BestLookup. To better understand the tradeoffs inherit in a marketplace design, we additionally evaluate a concrete implementation of a marketplace system, which we dub Video Delivery eXchange (VDX).

6 VDX IN DETAIL

We now discuss the design of our system VDX, which creates a *marketplace* where CDNs express performance and cost concerns at the cluster level, by programmatically sending bids to a broker (similar in spirit to an advertising exchange), to serve clients in specific locations. Brokers may use multiple bids both across and within CDNs to maximize the CPs' QoE and cost goals, while allowing CDNs to be paid appropriately. While this design is complex, it represents a plausible point in the design space with reasonable tradeoffs, although other valid choices exist as well. The large variation in CDN internal cluster cost (Figure 3), however, may warrant complex designs.

Below, we fill out the key details that define the instance of our Marketplace design, describe a few simple examples, and discuss some system challenges such as failures and fraud.

6.1 Decision Protocol Details

Share: Brokers send client (meta-)data to CDNs. The specific format may vary depending on the requirements of the marketplace; we use the simple format:
$[share\_id,\ location,\ isp,\ content\_id,\ data\_size,\ client\_count]$. Each share contains an opaque *share_id* for use in *Matching* and *Announce*. This format can easily be extended to include other meta-data, e.g., client device type, depending on CP's optimization goals.

Announce: CDNs send the output of *Matching* ("bids") to brokers for optimization. Similar to *Share*, the format of each bid should be specialized to the needs of the marketplace. We use the simple format:
$[cluster\_id,\ share\_id,\ performance\_estimate,\ capacity,\ price]$
Each bid includes a *cluster_id* (an opaque id known only between the broker and the CDN), a *share_id* from *Share*, performance estimates from *Estimate*, cluster capacity, and a price related to internal cost. CDNs may express policy by choosing not to bid on certain client announcements (e.g., certain videos cannot be served from certain CDN clusters, etc.), or by announcing modified performance, capacity, or price values. The broker trusts that these values are accurate. CDNs that continuously provide inaccurate values can be held accountable by lowering the priority of their bids, or by taking legal action if this goes against their contract.

Accept: Brokers communicate the results of *Optimize* to CDNs, including CDNs that "lost" the auction. This allows CDNs to understand which bids were accepted, allowing the CDN to prepare different bids (e.g., ones with lower prices, higher performance estimates, etc.) for the next round of bidding. Once again, the format should meet the needs of the exchange. We use the simple format:
$[cluster\_id,\ share\_id,\ performance\_estimate,\ capacity,\ price]$
The accept format is likely the same as the bid format.

6.2 Examples

VDX addresses many of the problems faced in §3. Three such fixes are: 1) VDX uses per-cluster pricing. This addresses the scenario in Figure 6. CDN X loses money as only its expensive cluster is used. With VDX's per-cluster pricing, CDN X can bid using its cheap clusters at a competitive price. 2) Traffic unpredictability (Figure 4) is greatly reduced in VDX as CDNs are explicitly involved before brokers move any traffic. 3) Applications with non-standard QoE metrics (e.g., latency agnostic applications) are easy to accommodate by having CDNs send bids that do not prioritize latency.

6.3 Challenges and Limitations

"Weak" traffic predictability: Although better than today's world, VDX's marketplace design only provides weak traffic predictability, as it runs a single phase of bidding. Traffic may move more quickly than some CDNs want, or a particularly bad set of bids may be accepted in tandem, leading to overloaded clusters. An

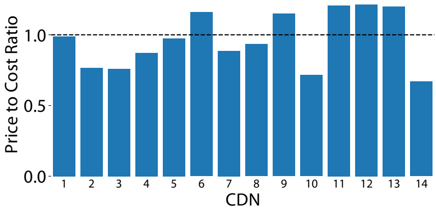

Figure 10: Per-CDN price to cost ratio for `Brokered` (less than 1.0 means profit loss).

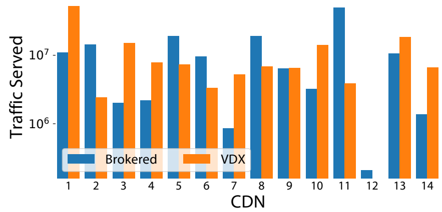

Figure 11: Per-CDN traffic for `Brokered` and `VDX`.

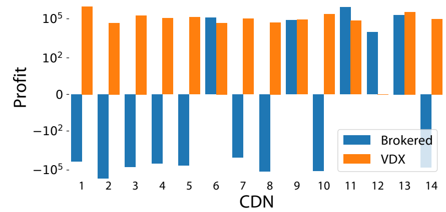

Figure 12: Per-CDN profits for `Brokered` and `VDX`.

optimal design would require all CDNs to agree on the broker's allocation (effectively multiparty consensus), which is impractical (§4). We argue instead that, in `VDX`, CDNs can learn risk-averse bidding strategies over time that will likely provide traffic predictability. Modeling these strategies with game theoretic frameworks (similar to those looking at CDN pricing [57], ISP transit pricing [58], or CDN-ISP collaboration [34, 35]) provides an interesting future research direction.

Failures and poor performance: With many different entities, Decision Protocol failures may seem difficult to combat. If a CDN has a failure, the rest of the system still continues to work. Failures or poor performance in the Delivery Protocol are handled using a variety of recovery mechanisms (e.g., moving clients mid-stream), as is done today. As brokers solely exist to optimize performance, when a broker fails, CP software can always fail gracefully to ignoring the broker and request content from a given CDN directly.

Fraud: CDNs that consistently send fraudulent bids (or fail often) can be marked as "bad" using a reputation system. Their bids can be handled at lower priority in the brokers' decision process. Brokers that fail to provide clients to CDNs can be handled similarly. CDNs that collude with other CDNs or brokers can be handled similarly, or through legal action if CDN-broker contracts become commonplace (see §8).

Scalability limitations: For scalability, instances of `VDX`'s marketplace would most likely need to focus on specific geographic regions, content providers, or content types. However, this division comes at a cost: limiting the broker's view limits the quality of the optimization. Federating these different marketplaces (as well as those run by different brokers) remains an open question.

7 EVALUATING A MARKETPLACE DESIGN

We evaluate `VDX` using the same simulation methodology from §5.1, focusing on comparing `Brokered` to the potential benefits of `VDX`'s `Marketplace` design, in three broad categories: 1) *data driven* which focuses on simulating `VDX` on real data, 2) *scenarios* which augment the data driven simulation, and 3) *microbenchmarks* which adjust knobs within `VDX` to look for trade-offs.

7.1 Data Driven

We answer the following two questions:

1. *How do different CDN deployment models compare? Does brokering really treat different CDNs differently?* `Brokered` makes it harder for distributed CDNs to make profits; `VDX` provides fairness.
2. *Do countries see pricing issues?* `Brokered` causes country-level pricing issues (some entirely unprofitable). `VDX` is cost-aware, moving traffic to cheaper ones, and charging appropriately.

7.1.1 CDN-level Pricing Differences. Here we examine how brokering today affects individual CDNs. In Figure 10, we show the ratio of flat-rate contract price to cost for `Brokered`. Recall that we compute contract price as an average over all clusters when the CDN is offered the entire workload. We markup this price by 20% to ease later comparison. If the price to cost ratio is less than 1.0, the CDN is losing money on delivery.

Most CDNs do not profit on *brokered video delivery* in our model of a flat-rate world, which may accurately represents the hardships present in some CDNs quarterly filings [6, 7, 42, 47, 48]. Video delivery is traditionally hard to profit on, given its high-bandwidth, low-"importance-per-bit" nature. While video delivery makes up a large portion of CDNs' cost [7], only a subset of it is brokered video delivery. The trend, however, towards using brokers for video delivery, even among small CPs, is rapidly accelerating [56], making these issues even more pertinent.

Examining the CDNs that make profits, we note that they are all centrally deployed CDNs mainly used in locations where costs are cheap. CDNs profit in `Brokered` only if they use clusters that are cheaper than their contract price. Today's world disincentivizes building large distributed CDNs, as distributed CDNs are more likely to be picked by brokers due to their better performance, yet their larger geographical presence potentially leads to higher cost variability.

Figure 11 shows traffic allocation across CDNs. Although CDN 12's cheap clusters are used by our broker, it does not actually serve much traffic. More distributed CDNs, such as CDN 1, have more variability in cluster cost as they are in many more remote regions (see Figure 3). Because of this, CDN 1 has an expensive flat-rate price (i.e., median cluster cost), so it is avoided by `Brokered` in favor of the comparably cheaper CDN 11. Moving to `VDX` allows CDN 1's prices to reflect individual cluster costs, allowing `VDX` to use CDN 1's cheaper clusters while avoiding its expensive ones.

Figure 12 crisply illustrates this switch. Here we plot each CDN's profits in `Brokered` and `VDX`. In `Brokered`, profit is a markup factor (1.2) times the contract price minus internal CDN cost. `VDX` uses the internal cost as the price, meaning profit is just the markup factor (1.2) times the cluster cost minus the cost. In `Brokered` many expensive CDN clusters are (unknowingly) used, leading to significant deficits for many CDNs in this flat-rate price model. `VDX`'s per-cluster cost model effectively levels competition, allowing each CDN to make profits, regardless of its deployment style.

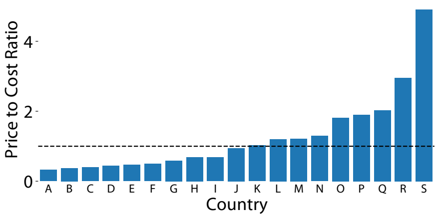

Figure 13: Per-country price to cost ratio for Brokered (less than 1.0 means profit loss).

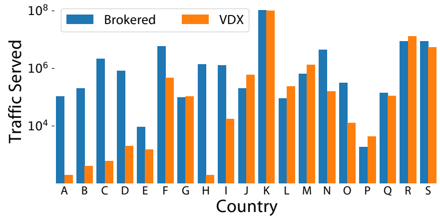

Figure 14: Per-country traffic for Brokered and VDX.

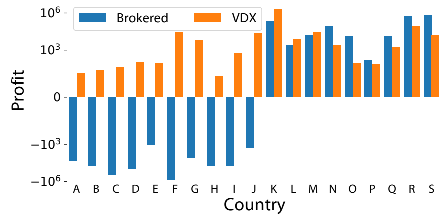

Figure 15: Per-country profits for Brokered and VDX.

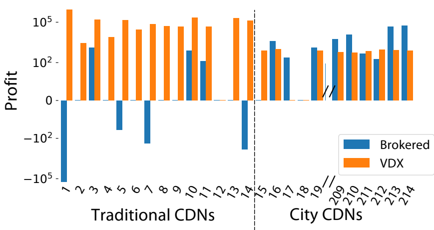

Figure 16: Profits for 200 "city-centric" CDNs added to our trace.

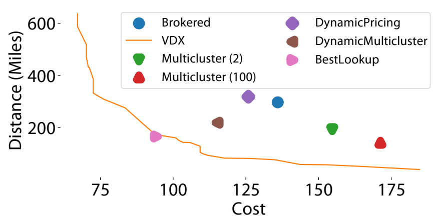

Figure 17: Adjusting optimization to balance performance vs. cost.

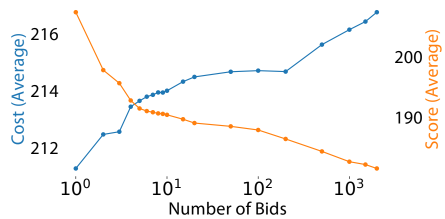

Figure 18: Adjusting bid counts vs. cost and score (lower is better).

7.1.2 Country-level Cost Differences. We examine the same data per-country. In Figure 13, we see that putting clusters in certain countries is more profitable; namely, countries L–S are easy to profit in, but countries A–J are where CDNs are losing money.

Interestingly we see different patterns in per-country traffic (Figure 14) than in per-CDN traffic (Figure 11). Country use is mostly even for `Brokered`. `VDX`, however, avoids the most expensive countries (A-E). This implies that `VDX` is sending traffic originating within these countries to clusters in cheaper countries. This may be reasonable in places like Europe, where neighboring countries are geographically close.

Figure 15 shows CDN profits, calculated similar to Figure 12. Here it becomes very clear that in `Brokered` CDNs in countries A–J are losing money, but with `VDX`, CDNs are able to profit even within these expensive countries, as CDNs can now be properly paid for using their expensive clusters.

7.2 Scenarios

Here, we answer the following question: *How do hundreds of "city-centric" CDNs affect established CDNs in today's brokered world?* "City-centric" CDNs are always profitable, while traditional CDNs lose money. `VDX` provides a fair playing field.

As previously explained, brokering (both what we see today, as well as our vision of a marketplace), allows for more varied, specialized CDNs. With brokering, CDNs no longer need to provide global coverage (as brokers can stitch together many smaller CDNs), allowing for a rise of "city-centric" CDNs. We model the advent of CDN proliferation by generating 200 single-cluster CDNs to add to our trace. Each cluster is drawn randomly from the CDN location data we collected from PeeringDB [53]. As these clusters are co-located with other CDNs, they drive down the co-location cost in our model.

We show the profits for these CDNs in Figure 16 (the city CDNs not shown have similar profits). We find that many traditional CDNs continue to do poorly in `Brokered` as they do in Figure 12, while some are sent no traffic at all, but the city CDNs always profit. This is because the cost of their single cluster is always equal to their contract price (as it is their average price), and thus they profit. `VDX` levels out the playing field, allowing traditional CDNs to properly compete.

7.3 Microbenchmarks

We answer the following two questions:

1. *How much control do CPs have over* `VDX`*'s cost / performance trade-off?* Points on `VDX`'s trade-off curve outperform most other designs.
2. *How much impact does CDN bid count have on performance and cost?* Bid count can improve performance, but generally has diminishing returns.

7.3.1 Understanding the performance / cost trade-off. In Figure 17 we vary the cost weight w_c in the optimization function run by our broker (see §4.2). Not only can `VDX` lower the cost by ~44% while keeping distance equivalent to `Brokered`, it can instead lower distance by ~74% while keeping cost equivalent. At the knee of the curve, it can simultaneously lower cost and distance by ~31% and ~40% respectively. There are similar trade-offs that can be made with most other designs.

7.3.2 Number of Bids. Here we vary the number of bids that CDNs submit for every client location. We show its effects on the average cost and score in Figure 18. As bids are sorted based on cost, increasing the number of bids should allow better performance (lower score) at higher cost. Interestingly, the largest increase in performance (drop in score) is just achieved by adding the second bid. Having two choices provides much benefit for brokers in meeting CPs goals, but as we have seen, having many more choices and tuning the trade-off is likely more important.

8 DISCUSSION

Adoption incentives: While CDNs have incentives to use dynamic cluster pricing (as it removes discrepancies between cluster price and cost), CPs may be hesitant to change their contracts. Similarly, while cluster-level optimization is incentivized for CPs/brokers (to better meet CP goals), CDNs may balk at the idea of providing brokers any additional control (although brokers already ultimately decide which CDN clients go to). We argue that requiring both (seen in a few designs in §4.2), provides enough incentive for both CDNs and CPs/brokers.

VDX's marketplace requires very little change to the existing "ecosystem," rather than the creation of an entirely new one (e.g., CDN federation, which inherently requires competitors working together). Furthermore, a marketplace design provides incentives to both large and small CDNs, as it allows both to compete on equal footing (§7.2). More nuanced CDN pricing schemes (e.g., low-but-variable pricing combined with high-but-flat pricing, similar to Amazon EC2 [8]) could offer CPs more control in meeting their goals, while retaining similarity to today's flat-rate pricing.

Lack of ISP integration: The lack of ISP integration is a purposeful limitation of this work. First, while there has been much spirited work looking into the ISP-CDN tussle [22, 34, 35, 54, 55], there has been little work focused on the CDN-broker tussle [45]. We view these works as orthogonal to ours, potentially fitting together into a single delivery ecosystem. Additionally, the lines between ISPs and CDNs are becoming much more blurred as large ISPs run their own CDNs (e.g., Comcast [11]), purchase CDN systems from vendors like Huawei [30] or Akamai [5, 59], or allow CDNs to run *virtual servers* within the ISP [21, 22].

Evolving the ecosystem: VDX's marketplace makes it much easier for CPs to meet their goals across a wide array of CDNs. In today's world, CPs sign contracts with CDNs directly, even if they use a broker. We do not need to assume this for VDX's marketplace. Similar to the evolution of online advertising networks, specific CP-CDN contracts could be removed to much more easily meet CP goals (by using many more CDNs), as well as lower the barrier of entry for new CDNs. If CP-CDN contracts are removed, we expect CP-Broker contracts and CDN-Broker contracts to replace them. Additional intermediary players (e.g., geographic CDN aggregators) may pop up in the ecosystem (similar to ad networks).

Extensions to non-video content: Although VDX is designed with video delivery in mind (as that is where brokers are seen today), there is nothing inherently video-specific about its marketplace. While the optimization done by a broker on behalf of the CP would need to be adjusted, we expect that VDX could be extended to cover different types of content and applications.

The true impact of cost savings: 31% bandwidth and co-location cost savings may seem small, but would save Akamai ~$22.7M per quarter [7]. While clients would also benefit from multiple cluster choices (decreasing cluster distance), the pressing issue is that many of the parties involved in video delivery are having difficulty making much money from it, with some losing money on it [47, 48] or experiencing slowing revenue growth [7, 42]. If video delivery could be assured to be profitable (§7.1), that is significantly more impactful than cost savings.

9 RELATED WORK

Collaboration in content delivery: The most relevant related work looks at widening interfaces in content delivery through collaboration. This includes alternative CDN designs, such as federated Telco-CDNs [12] and P2P-CDN hybrids [12, 65], and the potential benefits of CDN-ISP collaboration [22, 54, 55]. Some focus on the mathematical basis of joint collaboration [34, 35]. These works show that ISPs can aid CDNs in assigning clients to CDN clusters.

Experience Oriented Network Architecture (EONA) [32] argues abstractly that content owners and infrastructure owners should collaborate to improve end clients' QoE. Though similar, we focus on concrete problems faced by CDNs and brokers, and how to fix them.

Other collaboration proposals: Other work on ISP-P2P collaborations [15, 62] or ISP-ISP collaborations [41] are also related in terms of their designs. Both have an actor (an ISP) communicate a set of preferences (i.e., costs) over a set a set of resources (ISP paths), which are then chosen by another actor (an application / another ISP). Neither, however, treat this as a marketplace where bids change over time to strategically match performance / cost goals.

Route Bazaar [17] is more closely related in design. It allows customers to build end-to-end ISP paths using a marketplace. Tuangou [58] propose customer ISPs collaborate to share the cost of upstream provider service. While similar, neither are directly applicable to CDN-broker collaboration

Online marketplaces: We note strong parallels to online marketplaces, in particular those related to advertising. The most useful for our context are survey papers tracking the evolution from one-on-one contracts to ad networks (e.g., Google AdWords [26]) to ad exchanges (e.g., DoubleClick by Google [27]) [52, 60, 63]. Work in other networking domains have also decried flat-rate pricing in the context of inter-datacenter transfers [31]. Finally, different auction-style pricing mechanisms have been applied to cloud-computing [61, 64].

10 CONCLUSION

The introduction of brokers into CDN-based content delivery may have caused many issues for both CDNs and brokers due to the lack of explicit joint decision making. Using data from both a broker and a CDN, we show that: 1) brokers need cluster-level info to best meet CP goals, 2) CDNs are not being fairly paid due to the lack of cluster-level pricing, and 3) traffic patterns are unpredictable. We argue that there is a rich design space that solves all three problems, with a marketplace-inspired design providing potentially nice tradeoffs. We design a marketplace-based system called VDX that allows all CDNs to profit on video delivery, improving cost and data-path distance.

ACKNOWLEDGMENTS

The authors would like to thank Raja Sambasivan, Peter Steenkiste, Bala Chandrasekaran, KC Ng, Junchen Jiang, Yuchen Wu, David Naylor, our shepherd Ganesh Ananthanarayanan, and the anonymous reviewers for their help and feedback. This work is supported in part by NSF awards #CNS-1345305 and #CNS-1345284.

REFERENCES

[1] 2012. Cisco Report: The CDN Federation - Solutions for SPs and Content Providers To Scale a Great Customer Experience. http://www.cisco.com/c/dam/en_us/about/ac79/docs/sp/CDN-Federation_Phase-2-Pilot.pdf. (2012).

[2] 2015. Conviva. http://www.conviva.com. (2015).

[3] 2017. Level3 Communications. http://www.level3.com/en/products/content-delivery-network/. (2017). http://www.level3.com/en/products/content-delivery-network/

[4] Akamai. 2015. Financial Reports. http://www.ir.akamai.com/phoenix.zhtml?c=75943&p=irol-reportsannual. (2015).

[5] Akamai. 2017. Akamai. https://www.akamai.com/. (2017).

[6] Akamai. 2017. Akamai Reports First Quarter 2017 Financial Results. https://seekingalpha.com/pr/16818558-akamai-reports-first-quarter-2017-financial-results. (May 2017).

[7] Akamai. 2017. Akamai Technologies 2017 Q1 Earnings Summary. https://seekingalpha.com/filing/3541289. (May 2017).

[8] Amazon. Amazon Elastic Compute Cloude (Amazon EC2). http://aws.amazon.com/ec2/. (????).

[9] Amazon. 2016. Amazon CloudFront. https://aws.amazon.com/cloudfront/. (2016).

[10] Amazon. 2016. Amazon CloudFront Pricing. https://aws.amazon.com/cloudfront/pricing/. (2016).

[11] arstechnica. 2014. It's not a fast lane but Comcast built a CDN to charge for video delivery. http://arstechnica.com/information-technology/2014/05/its-not-a-fast-lane-but-comcast-built-a-cdn-to-charge-for-video-delivery/. (2014).

[12] Athula Balachandran, Vyas Sekar, Aditya Akella, and Srinivasan Seshan. 2013. Analyzing the Potential Benefits of CDN Augmentation Strategies for Internet Video Workloads. In *Proceedings of the 2013 Conference on Internet Measurement Conference (IMC '13)*. ACM, New York, NY, USA, 43–56.

[13] Athula Balachandran, Vyas Sekar, Aditya Akella, Srinivasan Seshan, Ion Stoica, and Hui Zhang. 2013. Developing a predictive model of quality of experience for internet video. In *SIGCOMM '13: Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*. ACM Request Permissions.

[14] Bizety. 2014. The Multi-CDN Strategy. https://www.bizety.com/2014/05/09/multi-cdn-strategy/. (2014).

[15] Eric W Burger and Jan Seedorf. 2009. Application-layer traffic optimization (ALTO) problem statement. (2009).

[16] Rajkumar Buyya, Mukaddim Pathan, and Athena Vakali. 2008. *Content delivery networks*. Vol. 9. Springer Science & Business Media.

[17] Ignacio Castro, Aurojit Panda, Barath Raghavan, Scott Shenker, and Sergey Gorinsky. 2015. Route Bazaar: Automatic Interdomain Contract Negotiation. In *15th Workshop on Hot Topics in Operating Systems (HotOS XV)*. USENIX Association, Kartause Ittingen, Switzerland. https://www.usenix.org/conference/hotos15/workshop-program/presentation/castro

[18] Cedexis. 2017. Cedexis. https://www.cedexis.com/. (2017).

[19] Fangfei Chen, Ramesh K. Sitaraman, and Marcelo Torres. 2015. End-User Mapping: Next Generation Request Routing for Content Delivery. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication (SIGCOMM '15)*. ACM, New York, NY, USA, 167–181. DOI:https://doi.org/10.1145/2785956.2787500

[20] CloudFlare. 2016. Bandwidth Costs Around the World. https://blog.cloudflare.com/bandwidth-costs-around-the-world/. (2016).

[21] Benjamin Frank, Ingmar Poese, Yin Lin, Georgios Smaragdakis, Anja Feldmann, Bruce Maggs, Jannis Rake, Steve Uhlig, and Rick Weber. 2013. Pushing CDN-ISP Collaboration to the Limit. *SIGCOMM Comput. Commun. Rev.* 43, 3 (July 2013), 34–44.

[22] Benjamin Frank, Ingmar Poese, Georgios Smaragdakis, Anja Feldmann, Bruce M Maggs, Steve Uhlig, Vinay Aggarwal, and Fabian Schneider. 2013. Collaboration opportunities for content delivery and network infrastructures. *Recent Advances in Networking* 1 (2013), 305–377.

[23] Aditya Ganjam, Faisal Siddiqui, Jibin Zhan, Xi Liu, Ion Stoica, Junchen Jiang, Vyas Sekar, and Hui Zhang. 2015. C3: Internet-Scale Control Plane for Video Quality Optimization. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*. USENIX Association, Oakland, CA, 131–144. https://www.usenix.org/conference/nsdi15/technical-sessions/presentation/ganjam

[24] Richard J Gibbens and Frank P Kelly. 1999. Resource pricing and the evolution of congestion control. *Automatica* 35, 12 (1999), 1969–1985.

[25] Google. 2016. Google Cloud CDN. https://cloud.google.com/cdn/. (2016).

[26] Google. 2017. AdWords. https://adwords.google.com/. (2017).

[27] Google. 2017. DoubleClick by Google. https://www.doubleclickbygoogle.com/. (2017).

[28] Gurobi. 2017. Gurobi Optimization. http://www.gurobi.com/. (2017).

[29] Dongsu Han, Robert Grandl, Aditya Akella, and Srinivasan Seshan. 2013. FCP: a flexible transport framework for accommodating diversity. *ACM SIGCOMM Computer Communication Review* 43, 4 (2013), 135–146.

[30] Huawei. 2017. Huawei. https://www.huawei.com/. (2017).

[31] Virajith Jalaparti, Ivan Bliznets, Srikanth Kandula, Brendan Lucier, and Ishai Menache. 2016. Dynamic Pricing and Traffic Engineering for Timely Inter-Datacenter Transfers. In *Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference*. ACM, 73–86.

[32] Junchen Jiang, Xi Liu, Vyas Sekar, Ion Stoica, and Hui Zhang. 2014. EONA: Experience-Oriented Network Architecture. In *HotNets-XIII: Proceedings of the 13th ACM Workshop on Hot Topics in Networks*. ACM Request Permissions.

[33] Junchen Jiang, Vyas Sekar, Henry Milner, Davis Shepherd, Ion Stoica, and Hui Zhang. 2016. CFA: A Practical Prediction System for Video QoE Optimization. In *Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation (NSDI'16)*. USENIX Association, Berkeley, CA, USA, 137–150. http://dl.acm.org/citation.cfm?id=2930611.2930621

[34] Wenjie Jiang, Rui Zhang-Shen, Jennifer Rexford, and Mung Chiang. 2008. Cooperative Content Distribution and Traffic Engineering. In *Proceedings of the 3rd International Workshop on Economics of Networked Systems (NetEcon '08)*. ACM, New York, NY, USA, 7–12. DOI:https://doi.org/10.1145/1403027.1403030

[35] Wenjie Jiang, Rui Zhang-Shen, Jennifer Rexford, and Mung Chiang. 2009. Cooperative Content Distribution and Traffic Engineering in an ISP Network. In *Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '09)*. ACM, New York, NY, USA, 239–250. DOI:https://doi.org/10.1145/1555349.1555377

[36] Frank P Kelly, Aman K Maulloo, and David KH Tan. 1998. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research society* 49, 3 (1998), 237–252.

[37] USC-Network Systems Lab. 2015. CDN Geographic Coverage. http://usc-nsl.github.io/cdn-coverage/. (2015).

[38] Philip A Lisiecki, Cosmos Nicolaou, and Kyle R Rose. 2009. Scalable, high performance and highly available distributed storage system for Internet content. (Nov. 24 2009). US Patent 7,624,169.

[39] Xi Liu, Florin Dobrian, Henry Milner, Junchen Jiang, Vyas Sekar, Ion Stoica, and Hui Zhang. 2012. A case for a coordinated internet video control plane. In *Proc. ACM SIGCOMM*. 359–370.

[40] Bruce M Maggs and Ramesh K Sitaraman. 2015. Algorithmic nuggets in content delivery. *ACM SIGCOMM Computer Communication Review* 45, 3 (2015), 52–66.

[41] Ratul Mahajan, David Wetherall, and Thomas E Anderson. 2007. Mutually Controlled Routing with Independent ISPs.. In *NSDI*.

[42] MarketWatch. 2015. Akamai Technologies profit falls, stock suffers. http://www.marketwatch.com/story/akamai-technologies-profit-falls-stock-suffers-2015-07-28. (July 2015).

[43] MaxCDN. 2016. Which CDN Pricing Model Costs Less. https://www.maxcdn.com/blog/cdn-pricing-models/. (2016).

[44] Microsoft. 2016. Content Delivery Network (CDN) Pricing. https://azure.microsoft.com/en-us/pricing/details/cdn/. (2016).

[45] Matthew K Mukerjee, Ilker Nadi Bozkurt, Bruce Maggs, Srinivasan Seshan, and Hui Zhang. 2016. The Impact of Brokers on the Future of Content Delivery. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*. ACM, 127–133.

[46] Netflix. 2016. Netflix Open Connect. https://openconnect.netflix.com/en/. (2016).

[47] Limelight Networks. 2015. 2015 Annual Report. http://investors.limelight.com/. (2015).

[48] Limelight Networks. 2017. 2017 First Quarterly. http://investors.limelight.com/. (2017).

[49] Ben Niven-Jenkins, Francois Le Fauchuer, and Nabil Bitar. 2012. RFC 6707: Content Distribution Network Interconnection (CDNI) Problem Statement. https://tools.ietf.org/html/rfc6707. (2012).

[50] NPAW. 2017. NicePeopleAtWork (NPAW). https://www.nicepeopleatwork.com/. (2017).

[51] Erik Nygren, Ramesh K Sitaraman, and Jennifer Sun. 2010. The akamai network: a platform for high-performance internet applications. *ACM SIGOPS Operating Systems Review* 44, 3 (2010), 2–19.

[52] OpenX. 2017. Ad Exchange vs. Ad Network: How Do They Compare? http://openx.com/whitepaper/ad-exchange-vs-ad-network-how-do-they-compare/. (2017).

[53] PeeringDB. 2017. PeeringDB. https://www.peeringdb.com/. (2017).

[54] Ingmar Poese, Benjamin Frank, Simon Knight, Niklas Semmler, and Georgios Smaragdakis. 2012. PaDIS emulator: an emulator to evaluate CDN-ISP collaboration. In *SIGCOMM '12: Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*. ACM.

[55] Ingmar Poese, Benjamin Frank, Georgios Smaragdakis, Steve Uhlig, Anja Feldmann, and Bruce Maggs. 2012. Enabling content-aware traffic engineering. *SIGCOMM Computer Communication Review* 42, 5 (Sept. 2012).

[56] Dan Rayburn. 2016. Current State Of The CDN Market: DIY, Pricing Trends, Competitive Dynamics. http://blog.streamingmedia.com/wp-content/uploads/2016/05/2016CDNSummit-Rayburn-Pricing.pdf. (2016).

[57] Yang Song, Arun Venkataramani, and Lixin Gao. 2013. On the CDN pricing game. In *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*. IEEE, 339–344.

[58] Rade Stanojevic, Ignacio Castro, and Sergey Gorinsky. 2011. CIPT: Using Tuangou to Reduce IP Transit Costs. In *Proceedings of the Seventh COnference on Emerging Networking EXperiments and Technologies (CoNEXT '11)*. ACM, New York, NY, USA, Article 17, 12 pages. DOI: https://doi.org/10.1145/2079296.2079313

[59] Volker Stocker, Georgios Smaragdakis, William Lehr, and Steven Bauer. 2017. The growing complexity of content delivery networks: Challenges and implications for the Internet ecosystem. *Telecommunications Policy* (2017).

[60] Brett Stone-Gross, Ryan Stevens, Apostolis Zarras, Richard Kemmerer, Chris Kruegel, and Giovanni Vigna. 2011. Understanding fraudulent activities in online ad exchanges. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. ACM, 279–294.

[61] Q. Wang, K. Ren, and X. Meng. 2012. When cloud meets eBay: Towards effective pricing for cloud computing. In *2012 Proceedings IEEE INFOCOM*. 936–944. DOI: https://doi.org/10.1109/INFCOM.2012.6195844

[62] Haiyong Xie, Y Richard Yang, Arvind Krishnamurthy, Yanbin Grace Liu, and Abraham Silberschatz. 2008. P4P: Provider portal for applications. *ACM SIGCOMM Computer Communication Review* 38, 4 (2008), 351–362.

[63] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. 2013. Real-time bidding for online advertising: measurement and analysis. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*. ACM, 3.

[64] H. Zhang, H. Jiang, B. Li, F. Liu, A. V. Vasilakos, and J. Liu. 2016. A Framework for Truthful Online Auctions in Cloud Computing with Heterogeneous User Demands. *IEEE Trans. Comput.* 65, 3 (March 2016), 805–818. DOI: https://doi.org/10.1109/TC.2015.2435784

[65] Mingchen Zhao, Paarijaat Aditya, Ang Chen, Yin Lin, Andreas Haeberlen, Peter Druschel, Bruce Maggs, Bill Wishon, and Miroslav Ponec. 2013. Peer-assisted content distribution in akamai netsession. In *Proceedings of the 2013 conference on Internet measurement conference*. ACM, 31–42.