A methodology for estimating interdomain Web traffic demand

Anja Feldmann, Nils Kammenhuber, Olaf Maennel
TU München
München, Germany
{anja,olafm,hirvi}@net.in.tum.de

Bruce Maggs
CMU/Akamai Technology
Pittsburgh, USA
bmm@cs.cmu.edu

Ravi Sundaram
Akamai Technology
Boston, USA
koods@lcs.mit.edu

April 16, 2003

Abstract

The variability of traffic as observed in the Internet and imposed on the distributed infrastructure is determined by many factors that are not well understood. To improve our understanding, we introduce an interdomain traffic model that can capture changes to content, user behavior, routing, and intra- and interdomain traffic. We propose to distinguish two levels of abstraction: (1) *publisher demand:* the volume of load originating from a Web publisher and destined to a client set (2) *Web traffic demand:* the volume of load originating from a specific Web publisher's server (i.e., an IP address) and destined to a client set. Web traffic demands explicitly consider the infrastructure of the publisher. Therefore they are appropriate for studying interdomain routing questions. Publisher demands are useful for studying user and/or publisher behavior.

This paper introduces for the first time a methodology for deriving a significant subset of such demands based on four key observations.

1. We observe that a sizable fraction of the bytes delivered to clients originate at publishers that utilize content distribution networks (CDNs).
2. We show that it is possible to obtain by extrapolation the *overall* traffic served by all publishers that utilize CDNs to *all* client sets. In particular, to estimate the traffic between a publisher

and all clients, we combine the logs from the CDN with an estimate of the fraction of the publisher demand that is served by the CDN.

3. While it cannot be presumed that each client set will access the same content, it is fair to presume that the fraction of demand served by the CDN on behalf of a publisher does not change dramatically from client set to client set. Therefore it is sufficient to estimate the fraction by "just" observing a few large and diverse client sets.
4. It is possible to map publishers to IP addresses with the help of the DNS system and information available from the interdomain routing system.

Using logs from Akamai, a major CDN, and two different client sets, we discuss our experiences in deriving the interdomain traffic demands, and present a preliminary analysis of the observed dynamics of the demands.

1 Introduction

Engineering the Internet is challenging due to its distributed nature. Each Internet Service Provider (ISP) is responsible for administering its part of the network. An ISP determines the flow of traffic within its piece of the network and may influence the flow of traffic in neighboring networks. But it does not have end-to-end control of the path from the source to the destination. Yet shifts in user behavior, publishing of new Web content, and/or deployment of new applications can cause major traffic shifts. Since the majority of traffic in the Internet travels across multiple administrative domains, no single ISP is capable of estimating the fluctuations in traffic volume exchanged between various hosts in the Internet.

Web site publishers are another party, besides the ISPs, that determine the flow of traffic in the Internet. By choosing a certain architecture or policy with regards to the organization of their content they are influencing where Web traffic demands are originating. By publishing new content or moving content, they may even control the behavior of the users. Indeed one could argue that they have at least as much control over the Web traffic exchanged between various parts of the Internet as the ISPs themself. In the early days of the Internet, most publishers were using a single machine. This has changed over the years, as the availability of content has become critical. More and more content providers are moving towards using a distributed infrastructure or at least a diversified infrastructure (e.g., multi-homed to multiple ISPs, in multiple cities). Thus, the impact of the infrastructure decisions made by the publishers has grown over the years. For example, a major content provider that combines a distributed infrastructure with reactive techniques for distributing load across its infrastructure can significantly impact the dynamics of interdomain traffic.

In summary, there are many factors that determine the variability of traffic demands as observed by an ISP among them:

- changes in user behavior
- changes in content
- changes in content location
- changes in the client to content direction policy
- changes in routing policy at some other ISP

Previous work has focused on either understanding user behavior [1, 2, 3, 4, 5, 6, 7, 8], or some aspects of changes in content [9, 10], or on the effects of these changes, e.g., in terms of the traffic demands [11, 12] imposed on a tier-one ISP [13, 14, 15, 16, 17], or in terms of poor end-to-end performance experienced by the users. The latter can be observed via active measurements of delay, loss, or throughput [18], or passive monitoring of individual routers and links [19, 20].

However, in order to determine the requirements on the interdomain routing system and on the infrastructure of each ISP, we need an understanding of the dynamics of the input, e.g., the changes in user behavior, the content, the content location and the client to content direction policy. These are the fundamentals without which we cannot expect to be able to define the requirements for the interdomain routing system, an open research question [21], or the policies according to which an ISP may adjust its interior and/or exterior routing policies [22, 23]. Even for the evaluation of how well/badly the current routing system, BGP, is doing (e.g., see [24, 25, 26]), we need these kinds of inputs. Nevertheless, it is not just the infrastructure engineering that would benefit from a better understanding of the interdomain traffic flows. An accurate view of the dynamics of the end-to-end traffic demands is crucial for debugging performance problems as well.

But what is an appropriate abstraction to capture the effects highlighted above? At one extreme, we could focus on individual source-destination pairs or some aggregation thereof. While providing much insight, this approach would lose all association with the publisher. An alternative, focusing on a single ISP, would not just ignore the content information, but also the interdomain aspect of the publisher demands. Rather, the natural representation, with regards to users accessing content, is as a *publisher-to-clients* volume: the amount of information that is flowing from a publisher to its clients. There is a crucial difference between publisher and Web site: The publisher is an abstraction and might be responsible for multiple Web sites that are distributed at various places in the Internet, but all under one administration. Hence, the above is a natural abstraction. But to study the effects on the Internet of any decisions, one needs an association of the abstract and the real world. Accordingly we consider instantiations of the publisher to clients volume: an *IP address to clients* volume which is a subset of the publisher-to-client volumes. One needs to consider IP addresses rather than prefixes since a site can be hosted on multiple unrelated addresses and a single address may host multiple sites. We refer to the first abstraction as *publisher demand* and the second abstraction as *Web traffic demand*.

Unfortunately, either one needs information from all clients or all publishers. Neither appears manageable. In this paper, we propose to take advantage of the 80/20 rule, which suggests focusing on the publishers that are providing significant Web traffic volume to their end users. But even identifying, e.g., the top 100 Web servers [27] (in terms of bytes served), is not trivial. Nor can one expect that all of the publishers would be willing or technically capable of estimating and providing traffic demands. On the other hand, a significant fraction of the publishers of these Web servers are utilizing content distribution networks (CDN) as surrogate servers. While the Web traffic demands that involve a CDN may not be the most interesting ones with regards to interdomain traffic demands (which they aim to avoid!) they provide us with the opportunity to estimate other publisher demands. Here the crucial observation is that one may expect that the ratio of bytes served by the CDN vs. bytes served by the publisher's infrastructure to be reasonable constant across client sets. If this is indeed the case, then it is possible to estimate all publisher and Web traffic demands involving the customers of a CDN based on log data from the CDN. To estimate the ratio of data serviced by the publisher infrastructure vs. the CDN, we propose to evaluate the access patterns of a large client population. Overall, we provide a methodology for populating parts of the demand model from logs collected at content distribution networks, measurements from two sizable user sets, and the DNS and routing systems. The last two data sets are needed in order to derive the Web traffic demands from the publisher demands.

Our analysis focuses on reasonably large time scales, on the order of tens of minutes to hours rather than minutes. This is the time scale at which traffic engineering tasks occur [28]. In addition we assume that different client populations retrieve information from the CDN and the original publisher at the same relative rate. This may not hold for all publishers, e.g., those that treat clients in the USA differently than those in Europe. Furthermore we assume that we can identify how a publisher is distributing its load across its infrastructure. These are all heuristics and more work is needed in order to check the validity of the above assumptions. Nevertheless, to the best of our knowledge there has been no comparable approach of estimating interdomain traffic conducted to date.

The remainder of this paper is organized as follows: In Section 2 we describe our ideas for estimating interdomain traffic demands. Section 3 discusses our methodology for realizing these ideas. A description of the individual data sets necessary for the instantiation of the demand models is given in Section 4. In Section 5 we present the initial results of analyzing the spatial and temporal properties of the traffic demands. Finally, in Section 6 we summarize our experience and suggest future research directions.

2 Interdomain traffic flows: idea

This section starts with a brief overview of the current structure of the Internet and the process of content delivery. Next, we motivate and introduce models for publisher demands and Web traffic demands. We then discuss how to use information from CDNs and detailed user site analysis to populate the demand models. Next we show how to enhance the abstractions to derive Web traffic demands.

2.1 Content delivery

Most users use the Internet to exchange information with other users or for accessing information that is available somewhere in the Internet. Content Delivery Networks (CDNs) (see, e.g., [29, 30, 31, 32, 33, 34, 35]) are designed to offload the load from the originating server and at the same time improve the performance to the user. Most CDNs have a large set of servers deployed throughout the Internet and cache the content of the original site publisher at these servers. Therefore another view of CDNs is that they provide reverse proxy services for their customers, the site publishers. In order to take advantage of their distributed infrastructure, requests for data are redirected to the “closest” cache server. If done intelligently this will reduce network load and therefore network congestion and response time. The CDNs differ in their approach of redirecting traffic. Some, such as Akamai [36], use DNS to translate the hostname of a page request into an IP address of an Akamai server. This translation may consider the location of the client, the location of the server, the connectivity of the client to the server, the load on the server, and other criteria such as performance measurements. Considering the load on the server achieves load balancing and redundancy across the CDN infrastructure.

An example that shows how the CDN infrastructure is embedded in the Internet architecture is shown in Figure 1. The Internet is divided into a collection of autonomous systems (ASs). Each AS is managed by an In-

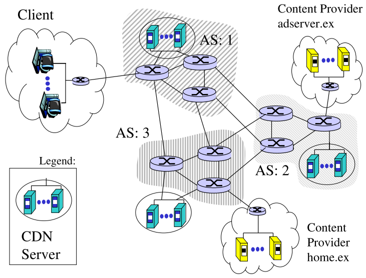


Figure 1: Example CDN deployment.

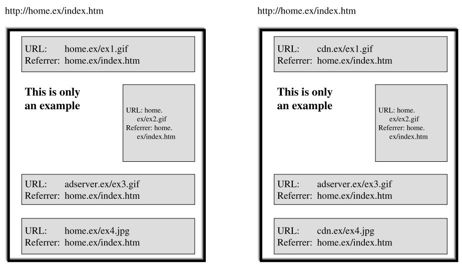
ternet Service Provider (ISP), who operates a backbone network that provides connectivity to customers and connects to other ISPs. In this case Figure 1 shows three ASs, numbered 1, 2, and 3, whose backbones consist of three routers each[1], two Web site publishers, `home.ex` and `adserver.ex`, and one set of clients. The site publisher `home.ex` is connected to AS 3 while the site publisher `adserver.ex` is connected to AS 2. A set of clients is connected to AS 1. Traffic is routed between the ASs by means of the Exterior Gateway Protocols [37], of which BGP [38] is the defacto standard. Traffic within an AS is routed by means of the Interior Gateway Protocols [39], e.g., OSPF [40], IS-IS, and RIP.

The location of the CDN's servers differ from CDN to CDN and depend on the contractual agreements between the CDNs and the individual ISPs. In some instances, the CDN's servers are deployed within the data centers of the ISP and therefore belong to the same AS, like AS 1 and AS 2 in Figure 1. Customers of the ISP (end users) will typically be served by these servers in the same AS. With other ISPs, the CDN may have a private peering agreement that allows the CDN to serve requests from the customers of the ISP via a direct connection between the CDN and the AS. The CDN may also co-locate servers with the ISP's customers, e.g., on university campuses. With other ISPs there may be no relationship with the CDN, and the traffic to the ISP's customers must be routed via another AS.

Let us consider the steps that are necessary to download the Web page shown in Figure 2 (a). This page consists of one main page located at `home.ex/index.htm` and four embedded objects. Three objects are located on the same server and one is served by, e.g., a company providing dynamic advertisements, `adserver.ex`. If a specific client from the client set connected to AS 1 in Figure 1 accesses the Web page, site publisher

---

[1] Most backbones consist of a larger number of routers, of course.



(a) no CDN (b) with CDN

Figure 2: Example Web pages.

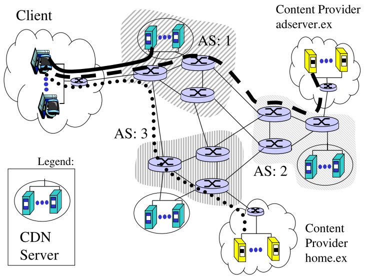


Figure 3: Example of traffic flows with a deployed CDN.

`home.ex` will serve the bytes for the main page and three embedded objects while site publisher `adserver.ex` will serve the bytes for the object located on its servers.

Now let us assume that site publisher `home.ex` has decided to use the services of a CDN. Note that a site publisher can decide not to have all of its pages hosted by the CDN, but only a part thereof. For example it determines that the chosen CDN should deliver two objects, `ex1.gif` and `ex4.jpg`, that are embedded in its home page, `http://home.ex/index.htm`. In this case, it has to mark the corresponding links, e.g., using ESI [41]. If a client from the same subnet now accesses this Web page, site publisher `home.ex` will serve only the bytes for the main page and one embedded object while the CDN will serve the bytes for the two embedded objects. The other site publisher `adserver.ex` is not affected by this policy change at `home.ex`. In terms of traffic flows, the CDN can service the traffic directly from its server farm in AS 1. This means that this traffic no longer has to traverse AS 3. Figure 3 shows the resulting traffic flows when downloading the Web page shown in Figure 2 (b).

2.2 Terminology

To simplify the discussion we briefly summarize some of the terms and abbreviations that we use in the remainder of the paper. The definitions are in part taken from the Web Characterization Terminology & Definitions Sheet [42].

Web site: A collection of interlinked Web pages, residing at the same network location.

Web site publisher: A person or corporate body that is the primary claimant to the rewards or benefits resulting from usage of the Web site and its content. A Web site publisher might publish multiple Web sites. Note that a site publisher is also often known as a content provider.

Subsite: A cluster of Web pages within a Web site that is maintained by a different publisher than that of the host site.

Supersite: A single, logical Web site that extends over multiple network locations, but is intended to be viewed as a single Web site.

Site infrastructure: A collection of servers on which a Web site (or a subsite) is hosted.

Physical Web site: A collection of "physical" Web sites, located at multiple host machines, that can be considered as one "logical" Web site, due to duplication and/or mirroring. The logical Web site can be identified using a unique domain name.

Content delivery network: An alternative infrastructure operated by an independent service provider on which some parts of a Web site can be hosted.

2.3 Demand Model

The interplay between content hosting, intra- and interdomain routing, and the Internet architecture is impacting our definition of traffic demands. In contrast to previous work by Feldmann et al. [13, 28], Medina et al. [14, 16, 15, 43], Xiao et al. [17] and Duffield and Grossglauser [44, 45] we are not focusing on a single ISP. Rather the goal of this study is the interdomain traffic imposed by all clients accessing content provided by many site publishers. In the following we first explain the concepts of publisher

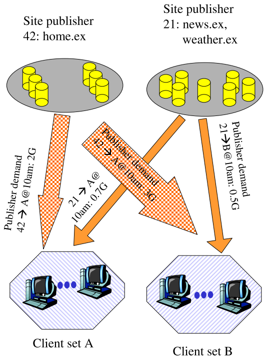


Figure 4: Publisher demands.

demand and Web traffic demands. We then discuss the motivation behind our choice for this abstraction from the viewpoint of routing, the clients, and the site publisher. We end with a brief review of the demand models.

Overview: In order to study changes to content and user behavior, we are mainly interested in aggregates at the level of the site publishers or client prefixes. On the other hand, in order to study routing we need to determine how much traffic is originating from each specific server that is hosting each site. Correspondingly we distinguish two different levels of aggregation and therefore two different kinds of traffic demands: *publisher demand* and *Web traffic demand*, see Figure 4 and 5.

Figure 4 shows two different publishers that are identified via some id (42 and 21) and the domain names of the sites that they publish: `home.ex` for 42 and `news.ex/weather.ex` for 21. Their content is accessed by two different client sets: A and B. Each client set accesses some of the content provided by `home.ex` and `news.ex/weather.ex` within the considered time period, e.g., starting at 10am and ending at noon. This results in traffic flowing from the Web sites of `home.ex` and `news.ex/weather.ex` to the client sets A and B. These traffic flows are what we refer to as *publisher demands*.

As already indicated in Figure 4 the infrastructure that is servicing the Web sites belonging to a publisher can be quite extensive and is likely to be distributed. To consider these effects, we introduce the concept of a *Web traffic demand*. A Web traffic demand shows the flow between a set of clients and a server on behalf of a specific publisher. Therefore the Web traffic demands are refinements of the publisher demands that also take

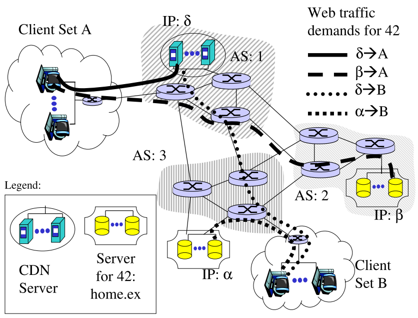


Figure 5: Web traffic demands.

into consideration the publisher's infrastructure. Figure 5 shows a simplified network topology with the distributed infrastructure for publisher 42. The Web site `home.ex` is hosted at servers within AS 2 and at servers connected directly to AS 3. The resulting four Web traffic demands are between client set A and B and IP addresses α, β, and δ. Some of the traffic comes from the to the CDN network since some of the Web pages (or objects appearing on Web pages) for `home.ex` may have been offloaded to the CDN as discussed above.

Routing: Interdomain routing operates at the granularity of network prefixes, and hence our Web demands are also defined in terms of network prefixes. Due to the large degree of connectivity in the Internet, there will most likely be many possible paths between any two prefixes. In particular, sites are often multi-homed and even some clients are multi-homed. Furthermore, most ISPs maintain peering or transit agreements with more than one other ISP. There may even be multiple links between neighboring ISPs. The ultimate decision [38, 37] as to which of the many possible routes for a destination prefix is chosen depends on the BGP route-selection process. While at each router BGP chooses a "shortest" path, this does not imply that the traffic between two prefixes will all follow the same path. Rather, traffic splitting between multiple different paths within an AS can be achieved using the capabilities of intradomain routing protocols, e.g., OSPF [40], which can split traffic along equal cost paths.

Clients: With regards to clients, summarizing them according to network prefixes is a good idea. Network prefixes provide a way of aggregating client traffic that preserves locality in terms of the Internet architecture.

Such an aggregation is necessary in order to avoid the severe scalability problems of representing each client at the level of an IP address. In addition, to some extent it addresses the problem of statistical significance due to too little traffic per individual IP address.

Site publisher: Summarizing site publishers via network prefixes is hazardous. A Web site publisher that serves a sizable fraction of all Web traffic is likely to use a distributed infrastructure. To influence the traffic flow, a publisher may use the DNS system. One way for multiplexing load between multiple servers is to return multiple resource records. By changing the order[2] in which the resource records are listed (round-robin), traffic will be split between the listed IP addresses. The second way is "proximity-aware" or "server-feedback dependent" resolution of the hostname. Here the authoritative DNS server may return different IP addresses in its answers. The IP address(es) it returns may depend on the location of the questioner, on the characteristics of the connectivity between the returned IP address and the IP address of the questioner, or on the load on the servers running on the host (or hosts) owning the IP address. For example if the client is located in Europe, the site publisher may redirect it to a server located in Europe and present a Web page in the local language (e.g., `google.com`).

Furthermore, it is sometimes impossible to deduce the Web site publisher from its IP address: A server may host multiple subsites of several site publishers. Even the name of a resource, e.g., the URL, does not in an obvious manner allow us to infer which site publisher is responsible for the content, e.g., `http://www.lordoftherings.com`. Some site publishers split their content into various sites, each with its own responsible organization and its own independent infrastructure. Hence it would be useful to keep the abstraction of a site publisher.

Demand model: To achieve our goal of aggregating clients, distinguishing site publishers, and considering routing, we use a two step approach, estimating *publisher demand* and estimating *Web traffic demand*. The difference is that the former will only capture the relationship between the clients and the site publisher, whereas the latter captures how the site publisher currently distributes its traffic. Accordingly, we associate each publisher demand with a *client prefix* that originates and/or receives the traffic, and a *Web site publisher* that serves the traffic. To map the publisher demands into Web traffic demands, we break the publisher's site infrastructure down to the IP addresses of its servers, and then map the

---

[2] Most clients use the first answer record.

demands from each client prefix to these IP addresses. The result is a Web traffic demand for each *client prefix* and a *set of IP addresses*, as shown in Figure 4 and Figure 5.

2.4 Demand Model: Applications

We now explore how our representation of traffic demands enables experimentation with changes to content hosting, to routing, to the AS level topology, as well as to the location of the content and/or the clients.

A publisher that needs to upgrade its infrastructure has many choices: upgrade the existing servers, add more servers, add more bandwidth on the existing network connection, add alternative network connections, change the way requests are allocated to individual servers, or outsource more of its content delivery. In order to decide on the best way to proceed, the publisher may use the publisher demands to evaluate possible scenarios: how the traffic volume is imposed by different client sets may influence his decisions. For his “what if” scenarios he just has to map the publisher demand to his potential infrastructure.

An ISP may also need to predict the effects that adding or moving a link or peering session may have. Based on the Web traffic demands it is possible to estimate which Web traffic from which publisher is affected. An important difference is that with Web traffic demands we know the traffic flows, not just through the network of the ISP, but throughout the Internet. Therefore it is easier to estimate what effect a certain decision, e.g., adding peering connections, will have. Furthermore it is possible to explore what effects policy changes will have. This is feasible not just for policy changes by the ISP itself but also for policy changes of other ISPs.

By combining Web traffic demands with topology and BGP routing information we can explore the impact of routing instabilities on actual traffic flows and vice versa. Furthermore by combining the Web traffic demands with performance measurements we can explore how user feedback needs to be factored into future considerations. Furthermore both the Web traffic demands as well as the publisher demands are ideal inputs for driving interdomain network simulations. As such our demand models have many different applications ranging from traffic engineering, to planing of content delivery, to network simulation.

2.5 Motivation for using CDNs to estimate publisher demands

Computing the publisher demands is possible given either information from each Web site publisher regarding which clients from which prefixes access the content served by the that publisher, or given information from each client set about which Web sites they are requesting. One way of deriving this information would be to collect fine-grain traffic measurements at all site publisher sites or all client sites. This may enable us to identify the traffic as it reaches the Web site publisher or the clients. However, this is not only extremely expensive but also virtually impossible since the huge number of site publishers/client sets imposes a task that is unmanageable. Furthermore it would still be necessary to address the question of how to distinguish site publishers co-located at a server.

Since the number of Web site publishers is smaller than the number of clients, we suggest focusing on the publishers. But instead of considering all publishers, we propose to take advantage of the fact that CDNs are in some sense (Section 2.1) providing reverse proxy services for their customers (the publishers), and are acting as their “subcontractors”. Using data collected within CDNs has several advantages:

- CDNs serve the content on behalf of their customers. This implies that the CDN has a way of relating content to customers.
- Due to the requirements imposed by volume based billing, CDNs collect data on behalf of their customers regarding how much traffic is served. This implies that the CDN has a way of deducing the amount of traffic it serves on behalf of a site publisher.
- In addition, most site publishers do not want to lose access to the information that they can collect when they serve content directly to clients – for example information about which clients are accessing what content is derivable from Web server logs. Accordingly the CDN has to collect this “Web server”-like log information. As a consequence, it has a way of relating traffic to clients.

Moreover the number of CDN service providers is significantly smaller than the number of publishers. For example a list of CDN organization types and their products is maintained by Teeuw [46] and by Davison [47]. To further reduce the candidate set, we observe that the market is dominated by only a small number of service providers such as Akamai [36], Cable & Wireless [48], and Mirror Image [49].

We note that focusing on CDNs limits us in terms of the number and kind of publisher demands that can be estimated. If a site publisher has no association with a CDN, it will not be possible to derive the publisher demands for this service. This raises the question of which publisher demands are we interested in and are those likely to be associated with a CDN. Like a lot of other quantities in networking [1, 50, 51, 13] and elsewhere [52] we expect publisher demands to be consistent with a Zipf-like distribution. A Zipf-like distribution is one where the contribution of the k-th most popular item varies as $1/k^a$, for some a. Since the heavy hitters will account for a significant part of the traffic we are mainly interested in them. Luckily those are the ones that are more likely to use the services of a CDN. Therefore CDNs can provide us with a way of estimating the publisher demands for the service providers that are most popular and thus account for a large part of the traffic.

Still one problem remains: as discussed in Section 2.1 and as shown in Figure 3, CDNs try to take advantage of their distributed infrastructure by serving traffic locally. Thus, how can we expect to derive estimates for interdomain Web traffic demands from traffic to CDNs? It turns out that most site publishers will not serve their whole content via the CDN. Rather they will use some mixture as for example shown in Figure 2 (b). Note that not all content has to be served via the Web site of the site publisher or the CDN. Rather some embedded objects may be located on yet another server, e.g., for the purpose of advertisement. This provides us with the opportunity that we need. If we know the ratio of a customer's traffic serviced via a CDN vs. via the servers of the publisher vs. via external sites, and if we can estimate one of these amounts, we also can estimate the other amounts. Most importantly we can do so for *all* client prefixes and *all* publishers that are customers of the CDN.

2.6 Extrapolating from CDN traffic to publisher demands: ideas

The above discussion leaves us with the problem of estimating the relationship of traffic between the various flows shown in Figure 3. One way to proceed could be to explore offline the content provided by the Web site of the publisher. Given a set of Web pages one can easily calculate the fractions of data served by the CDN vs. the fraction of data served by the original Web site. The problem with this approach is that it ignores the fact that certain Web pages are more popular than others.

Hence, we really need access to information about user accesses. There are many ways of doing this [53]: from users running modified browsers [1];

from the logs of the site publishers themselves [7, 54, 55, 56]; from proxies logging information about which data is requested by the users of the proxy [57, 58, 2]; or from the wire via packet monitoring [59, 60, 61, 62, 63]. Each of these methods has its advantages and most have severe limitations regarding the detail of information that they log. Distributing modified Web browsers suffers from access to the browser software and from users not accepting the modified browsers. While a few site publishers might cooperate by revealing their logs, most will not. In addition, this approach suffers from a scalability problem. Using proxy logs or logs derived via packet monitoring is more scalable with regards to service providers. But with regards to the size of the user population that can be monitored, it is more limited.

To choose the appropriate solution let us consider the granularity at which we need the information. The purpose of estimating the publisher demands is mainly to understand their medium time-scale fluctuations and their impact on traffic engineering, routing, etc. We are not as interested in small time-scale events (and in any case it is hard to understand their causes). Therefore some coarse grain estimation is sufficient for our purposes. Subsequently we propose a two-fold approach:

- to take advantage of the relationship between the CDN and their customers and acquire some subset regarding their estimate of the fraction of traffic that is served by the CDN and other third party providers.

- to use proxy and/or packet level traces to derive the fractions for some users and therefore for some sample client sets. (Either data source will do, since we are not interested in the details about, e.g., the protocol interactions, or the exact packets, or the exact temporal distribution, but rather we only want to estimate the relationships between traffic served by the CDN and traffic served by the Web site.)

2.7 From publisher demands to Web traffic demands

In order to derive the Web traffic demands from the publisher demands, we first need to map the Web sites of the publishers to IP addresses. This mapping may not be one-to-one because the publisher may use a distributed infrastructure and may use DNS for “load balancing” or “proximity-aware” or “server-feedback dependent” name resolution (see Section 2.3). Again, we propose to take advantage of information available to the CDN. It knows the set of hostnames that is associated with each site publisher. Therefore the problem is reduced to associating each hostname with its set of IP addresses.

This can be done using DNS queries. To account for "proximity-aware" or "server-feedback dependent" policies used by the site publisher, it is not sufficient to issue DNS queries from a single point in the Internet. Rather we need to use a set of DNS servers that are distributed throughout the Internet. Furthermore, we have to issue recursive queries[3] to these servers in order to discover their view of the server IP addresses. Thus, the DNS servers have to allow recursive DNS queries.

In a second step, we need to determine which server is used by which client. This problem can either be extremely simple or extremely hard. If the site uses a single IP address or simple DNS round robin across a number of different IP addresses this step is trivial. Since DNS round robin is supposed to partition the requests about evenly across all of the servers, this is what we will do in estimating demand. If the site uses a more sophisticated mechanism, we are left with a much harder problem. Here we have two possible ways to approximate the decision of the physical Web site. We can either use the result of the DNS server "closest" to the client set or we can assume that the client set is directed to the "closest" server. Here we propose to capture the meaning of "close" in terms of AS distance. This seems reasonable, since other measures of closeness are even harder to define, e.g., performance, and since it is known that some distributed infrastructures are using this information [64].

3 Estimating Web traffic demands

In the previous section we discussed the basic ideas regarding how to estimate publisher demands and Web traffic demands from information available at CDNs. In this section we present one more level of detail on how to estimate the various necessary pieces of information using logs from a CDN provider, packet-level measurements at ingress links, and the DNS system.

---

[3] In an iterative query the contacted name server tells the requesting name server which name server to ask next, while in a recursive query the contacted name server proceeds by sending a query to the next name server on behalf of the original user.

3.1 CDN log evaluation

To compute publisher demands using CDNs, fine-grain access records from all servers of the CDN should be collected. This can, for example, be realized via extended server log files collected on the CDN servers. The server should generate a record summarizing each transaction. Such records should be exported on a regular basis and should include sufficient information for computing the publisher demand: the `accessed resource`, the `client` IP address, the `start` and `end` times of the transfer, and the number of transferred `bytes`. (Any additional information could be used to further refine the notion of publisher demands.)

Computing the traffic demands requires information about the CDN customer (i.e., publisher) associated with each record. This aggregation process draws on a map, `resource_to_customerid`, such that every `resource` can be associated with a unique `customerid`. Furthermore, it uses another map, `clientip_to_clientprefix`, of network addresses such that every source IP address, `client`, can be associated with a network prefix `client_prefix`. The first map can be derived from the customer information of the CDN while the second can be derived with longest prefix match from a joined BGP routing table `joined_bgp_table` from multiple different viewpoints in the Internet.

No content transfer is instantaneous. Rather, they last for some time interval starting at `start`, ending at `end`, and contributing some amount of traffic, `transferred bytes`. In order to avoid problems in time resolution, e.g., discrepancies between clocks at the record collectors, granularity of the data sources, etc., and since most applications making use of publisher demands are on a larger time scale, we compute the demands on time scales of hours rather than minutes or seconds. Time is partitioned in bins of duration `bin_length`, according to the considered resolution. If a records spans multiple bins, we subdivide the traffic in proportion to the fraction of time spent in each time period.

To derive the final publisher demands we draw on two other maps, `customerid_to_demand` and `customerid_to_ external_demand`. The first one specifies for each customerid the relationship between the CDN hosted traffic flows and the self hosted traffic. The second one specifies the same information for each external server that the customer uses. The algorithm for computing the publisher demands in summarized in Figure 6.

```
For each accessed_resource: (client, start, end, transferred_bytes)
    customerid = resource_to_customerid(accessed_resource);
    clientprefix = longest_prefix_match(client, joined_bgp_table);
    start_bin = ⌊start/bin_length⌋ * bin_length;
    end_bin = ⌊end/bin_length⌋ * bin_length;
    if (start_bin == end_bin)
        volume[clientprefix, customerid, start_bin]
                += transferred_bytes;
    else /* Compute volume of traffic for each time_ bin */
        byte_rate = transferred_bytes/(end - start);
        volume[clientprefix, customerid, start_bin]
                += byte_rate * (start_bin + bin_length - start);
        for (time_bin = start_bin + bin_length; time_bin < end_bin;
                time_bin += bin_length)
            volume[clientprefix, customerid, start_bin] += byte_rate * width;
        volume[clientprefix, customerid, end_bin] += byte_rate * (end - end_bin);
Output for each aggregate: (clientprefix, customerid, time_bin, volume)
```

Figure 6: Estimating partial CDN publisher demands from CDN transaction logs.

3.2 Estimating flow ratios between CDN and Web publisher

In Section 2.6 we propose using proxy and/or packet level traces for estimating the relationships between the various flows shown in Figure 3. For this we take advantage of the relationships between the accessed objects. Consider again the example shown in Figure 2 (b). A log file, derived from the proxy log or the packet traces, should show six entries, one for each object (unless it is cached in the users cache). Each entry includes a unique `object_id`, the `url`, the `start` and `end` time of the download of the object, the transferred bytes (`trans_bytes`), and the `referer`[4] field (if specified by the user agent). Note that the referrer field, which lets a user agent include the URL of the resource from which the requested object was obtained, is optional and not necessary. Nevertheless most popular Web clients, such as Internet Explorer and Netscape, include them regularly. They prove to be extremely helpful. For example in our sample page, Figure 2 (b), all embedded objects have the same value for their referrer field independent of where the object actually resides. Indeed the value is the same as the `url` of the base page. Thus the referrer field provides us with the means to associate the objects and therefore provide us with the means of estimating the ratios between the traffic flows.

One way of estimating the ratios would be to try to compute the exact

---

[4] In the HTTP context, this is usually written “refe**r**er” and not “refe**rr**er” for historical reasons

temporal and causal relationship between the pages and their embedded objects. But past work, e.g., in the context of estimating the benefits of prefetching [58] or piggybacked cache validation [56, 53], has shown that this is a nontrivial task, especially in the presence of proxies and other strange users. But for our purpose we do not need the exact temporal relationship. Rather we are only interested in the fact that there *is* a relationship. Therefore we propose to use a much coarser three-pass approach[5].

The first two passes serve preparative purposes. In the first pass we separate the set of accessed objects according to users IP addresses. In the second pass, see Figure 7, we determine the set of objects served by the CDN under consideration, `cdn_set`, and some additional information that we specify below. For this purpose we check each object against the appropriate CDN customer base information `determine_customer_id()` and, if appropriate, compute the CDN `customerid` and add it to the `cdn_set`.

In the third pass we compute for each CDN object `cdn_id` within this set the possible base pages `base_candidate_set` and the possible other embedded objects `embedded_candidate_set`. For an object to fall into these sets either its URL or its referer has to be equal to the referer value of the CDN object. For this purpose we stored some additional information in the second pass: each object with URL `url` and referrer `referer` is added to the set of possible home pages for this URL `base_set(url)`. Furthermore, we add the object to the set of possible embedded objects for the current referrer `embedded_set(referer)`. Once we have retrieved the candidate sets, we can determine the hostnames for each of the objects within the candidate sets and add the bytes in the corresponding object to the appropriate traffic flow. The appropriate traffic flow is either determined by the `cdn_customer_id` for CDN objects or the hostname for non-CDN objects. If the hostname is not used in the users request, we propose to use the server IP address instead. In order to keep the relationship information, we can now establish the link `associated_hosts` between `cdn_customer_id` and the hostname of the objects in the candidate sets. In order to avoid double counting, e.g., if the exact same page is accessed multiple times, one needs to mark every object that has already been accounted for.

Again it is the case that no content transfer is instantaneous, but rather than spreading the contribution of each transfer across multiple time periods of duration `bin_length`, we propose to just add it to the last bin. It

---

[5] This three pass approach automatically ensures that Web pages referring to other Web pages are handled appropriately.

```
Pass 1:
Sort the accessed objects according to user IP addresses
Pass 2:
For each user IP and object_id: (url, start, end, trans_bytes, referer, hostname)
    if (determine_customer_id(object_id) evaluates to CDN object) then {
        customerid[object_id] = determine_customer_id(object_id);
        cdn_set ∪= object_id;
    }
    base_candidate_set[url] ∪= object_id;
    embedded_candidate_set[url] ∪= object_id;
Pass 3:
For each object_id from cdn_set with (url, start, end, trans_bytes, referer, hostname)
    if (done[object_id]) then next;
    done[object_id] = true;
    end_bin_cdn = $\lfloor end/bin\_length \rfloor$ * bin_length;
    cdn_customer_id = customerid[object_id];
    volume[cdn_customer_id, end_bin_cdn] ∪= trans_bytes;
    foreach candidate in (base_candidate_set[referer]
                or embedded_candidate_set[referer]) {
        if ($\exists$ customerid[candidate] or done[candidate]) then next;
        done[candidate] = true;
        associated_hosts[cdn_customer_id] ∪= hostname[candidate]
        end_bin_candidate = $\lfloor end[candidate]/bin\_length \rfloor$ * bin_length;
        volume_related[cdn_customer_id, hostname[candidate],end_bin_candidate]
                ∪= trans_bytes;
    }
Output for each customerid and host from the associated_hosts the ratios:
    (customerid, hostname, time_bin, volume[customerid, time_bin],
    volume_related[host, time_bin]/volume[customerid, time_bin])
```

Figure 7: Computing flow ratios: CDN vs. Publisher from user access logs.

is known [65] from aggregating Netflow [66] data that this can lead to artifacts. But if the aggregation periods are long enough, size and impact of these artifacts decrease significantly.

3.3 Mapping publisher demands to Web traffic demands

In order to map the publisher demands to Web traffic demands we need to find out which IP addresses are actually in use by the publisher's infrastructure. As an initial step, we derive the set of hostnames associated with each site publisher (`customer_id`) (via the mapping `customerid_to_hostname`), utilizing the knowledge of the CDN provider. Therefore the problem is reduced to associating each hostname (`host`) with its set of IP addresses (`ip_set`).

To account for the distributed infrastructure of the site we have to issue recursive DNS queries from a set of DNS servers distributed throughout the Internet. We propose identifying a set of candidate DNS servers from traffic measurements, such as Netflow [66] or packet level traces [67]. Using packet traces has the advantage that its easy to check if the DNS servers support recursive DNS queries. Otherwise one can issue a recursive query to the DNS server and see if it is willing to respond to the query and second if it supports recursive queries. Once we have derived a candidate set of DNS servers, we can either use all of them or a subset. We propose to concentrate on a subset such that each DNS server in the subset will return a different IP address for at least one Web site publisher that utilizes a distributed infrastructure. Since the CDN runs a highly distributed infrastructure we use the main Web server of the CDN, `www.cdn.ex`, for this purpose.

The next step involves identifying what kind of access distribution mechanism (`dns_policy`) is used by the physical Web site. We propose to concentrate on the popular mechanisms and look for indications of their use. If all queried DNS servers return almost [6] the same set of IP addresses then we can assume that DNS `round_robin` is used. If different DNS servers return different IP addresses in a consistent fashion (at least two times) then we can assume that some form of proximity-aware load balancing is used (`proximity`). In the first case we propose to split the load evenly between all IP addresses used to implement the physical infrastructure. Otherwise we propose to split the traffic only between the IP addresses resolved by the closest DNS server queried to the users in question. All other cases are currently resolved via manual inspection.

4 Data sets

The computation of the demands draws on several different data sets, as summarized in Figure 9 and 10. This section describes our approach for harvesting and preparing these various large data sets, each collected at a different location at a different granularity.

From the CDN: Using logs that feed into the CDN billing system of a major CDN provider we extracted the information which clients are accessing which information. Each individual log file records all accesses to some part of the CDN infrastructure during some time period and is available for processing some time after the last recorded access. We captured logs

---

[6] We use almost here since we cannot query all DNS servers at the same time and some temporal event can cause anomalies.

```
For each customer_id:
    hostname_set = customerid_to_hostname(customer_id);
    foreach host in (hostname_set) {
        foreach dns_server in (dns_server_set) {
            ip_set[customer_id] ∪= dns_query(dns_server, host);
            ip_set_dns[customer_id, dns_server] ∪= dns_query(dns_server, host);
            dns_policy[customer_id] = classify_dns_policy(ip_set)
        }
    }
Foreach client_prefix:
    closest_dns_server[client_prefix] = closest(client_prefix, dns_server_set);
Foreach customer_id and client_prefix:
    if (dns_policy[customer_id] == "round robin")
        split traffic evenly among ip_set[customer_id]
    if (dns_policy[customer_id] != "round robin")
        split traffic evenly among
            ip_set_dns[customer_id, closest_dns_server[client_prefix]]
```

Figure 8: Mapping site publishers to Web traffic demands.

Dataset	Location	Key Fields
CDN sites	CDN	List of Web sites and Web site publisher that use the CDN
CDN servers	CDN	List of physical Web sites
CDN logs	CDN billing system	Per accessed object: client IP address, resource, start and end time, transferred bytes
HTTP logs	external network connection	Per accessed object: user IP address, url, start and end time, transferred bytes, referrer, hostname
DNS lookups	set of name servers	Per hostname and DNS server: set of IP addresses
BGP table	peering points	Per network: set of possible routes (AS-path)

Figure 9: Datasets and key fields used in computing and validating the publisher and content traffic demands

for a roughly one hour time period around noon on Feb. 4th 2003 from more than 99.3% of all the operational servers of the CDN. There are two reasons why we did not capture logs from all servers: Logs for a certain time period arrive in bursts imposing a huge instantaneous burst overloading our collection infrastructure. Other logs, maybe delayed due to remote network outages, arrived after we stopped our data collection process. We initially aggregated this data using the methodology described in Figure 6 using a time aggregation of an hour. This time aggregation was chosen to examine the spatial rather than the temporal variability of the data.

From two user sets: Two sets of user access information were extracted from packet level traces from the external Internet connection of an university (UNI) and an organization (EDU) that provides Internet access to two major universities, several colleges and several research institutes. The capacity of the first one is 155 Mbit while the latter one has a capacity of 622 Mbit. In both locations the monitoring is realized via the monitoring port of a Gigabit Ethernet switch just before the traffic passes the last router to the Internet. We processed the raw packet stream on the fly to extract our final trace using the HTTP analyzer [68] of the intrusion detection system `bro` [69]. The resulting trace contains all relevant information and is much more compact than the raw packet data. The trace among other information records the fields shown in Figure 9.

Unfortunately due to the high traffic volume and since extracting HTTP logs from packet level traces is quite compute intensive [59], it is not possible to monitor the traffic from all internal machines/external Web servers to external Web servers/internal machines at the same time without significant packet losses. Therefore we partitioned the internal IP address space in 4/10 subsets for the monitoring points UNI/EDU such that the number of packet losses was reduced to a tolerable degree. At the first/second location the maximum observed packet loss rate within each 10 second time period was less than 0.5%/0.8%. Overall the number of 10 second time periods during the week with loss rates larger than 0.01% is less than 20/200. This implies that 99.97/99.67% of the time periods had a loss rates lower than 0.01%. To cover the full address space we monitored each subnet for approximately 3/2 hours at a time.

From the DNS system: We started by identifying roughly 7.000 DNS servers using packet level traces. The DNS server selection process ensures us that each server supports recursive DNS queries. But the process does not pay attention to the distribution of the DNS servers within the Internet infrastructure. Therefore in the next step we identified a subset of 467 DNS servers that return different results when resolving the name of the main CDN Web server. The 467 DNS servers are located in 427 ASs. We restrict ourself to using this subset in order to reduce the load on the overall DNS system while achieving a good coverage of the Internet infrastructure.

To resolve which of the publishers are using a distributed infrastructure, we selected a subset of 8.042 hostnames used by the publishers. The resolution of these hostnames resulted in more than 3.5 million queries. Of these, 93.4% returned a valid response. Overall these hostnames map to 9.164 hostnames of which 872 are operated by the CDN itself.

From the Routing system: We constructed a joined BGP routing table

Dataset	Date	Duration	Size
CDN logs	2/4/03		37 Gbyte
HTTP logs	1/22/03 - 1/29/03	1 week	2 and 2.2 Gbyte
DNS lookups	2/1/03 - 2/2/03	1 day	3.868K queries
BGP tables	11/29/02 & 2/4/03	-	27 tables each

Figure 10: Information for each data set

from the individual BGP tables on the 2/4/03 from the RouteView project [70]. In total these tables contained roughly three million entries. This table contained 127.104 routable entries. Furthermore we used the BGP tables from RouteView as well as Ripe's RIS project [71] to extract an approximation of the contractual relationships between the AS using a similar methodology as proposed by Gao [72].

5 Experimental results

In this section, we present our initial results of analyzing the various data sets discussed in Section 4 with regards to their characteristics and the steps for computing the publisher and Web traffic demands.

5.1 Estimating partial CDN publisher demands

The first step is estimating how much traffic is sent on behalf of each publisher by the CDN to each client set. We start with examining some basic statistics regarding the data sets. We observed 710.330 different client sets within the hour of the duration of data set CDN. This corresponds to a 23.6% coverage of the overall IPv4 address space and 52% coverage of prefixes within the routable IPv4 address space. 1.3% of the observed client space are not publicly routable. This may be due to placement of CDN servers within private networks. In total the client sets accessed roughly 3.6 Terabytes of data via the CDN network. This implies that each client set accessed about 5 Mbytes during this time period.

The Internet has obviously many client sets and a sizable number of publishers. But who is contributing the majority of the traffic. Is it a small set of client sets or a small subset of the publishers. Even by just studying the amount of traffic serviced by the CDN we can get a first impression of these relationships. In Figure 11 (a), we rank client sets by total traffic received from the CDN from largest to smallest, and plot the percentage of the total traffic attributable to each. As predicted we find a linear relationship on the log-log scale, an indication that the distribution is consistent with the characteristics of a Zipf-like distribution [52, 50]. But is this the case even for individual publishers? In Figure 11 (b) we explore the characteristics of the top 10 publishers, selected by the total number of bytes that they serve to all client sets. The fact that we still observe a linear relationship (for all but one) indicates that even the demands by a single publisher are dominated by the behavior of a few client sets. One aspect that may be contributing to these effects is that client sets are located in different time zones. About 36.4% of the client sets are located in the US, 16.9%*inJapan*,6.3% in Germany, and 4.7% in the UK.[7] One reason for a reduced amount of traffic is that for some networks most users are sleeping. Users of other client sets may be working, etc. For further evaluation of the user behavior and the impact on the traffic over time we plan to explore an even larger dataset in the future. As a first step we have classified each client set according to the time zone that it belongs to. We still observe popularity drops that are consistent with the Zipf-like distributions even if we just consider the traffic served to clients within a time zone.

Notice that Figure 11 (b) also shows that a client set that receives the most bytes from one publisher does not do so from another publisher. Rather there are a sizable number of fluctuations. This indicates that each publisher in the Internet has to determine for itself how the heavy hitters (contributers) are. Extrapolating from one set to another purely based on volume might be misleading.

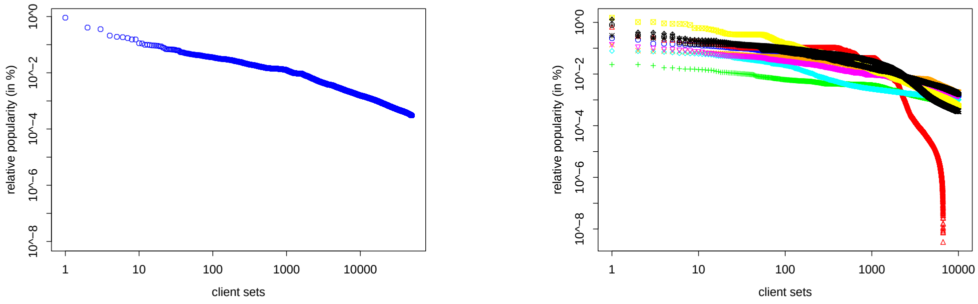
But what is the behavior if we consider the data from the view point of the client sets. In Figure 12 we explore the popularity of content served by the CDN on behalf the publishers. Again we observe a curve that indicates a Zipf-like distribution.

5.2 Estimating relationships between CDN and publisher flows

Once we know how much Web traffic is flowing from the CDN to each client set, we need some factor to extrapolate from the partial CDN publisher demands towards the full publisher demands. Accordingly we apply the methodology outlined in Figure 7 to the user access logs. Note that we are not necessarily capturing all of the traffic from the publisher since

---

[7] We used a product of the CDN to map the IP addresses to countries.



(a) All

top 10 publishers

Figure 11: (a) Percent bytes served by all publishers to client sets, listed in decreasing order. (b) Percent bytes served by top 10 publishers to client sets, listed in decreasing order.

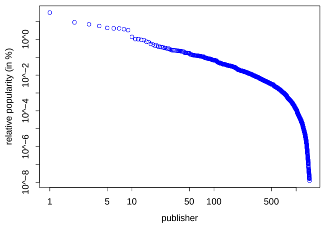


Figure 12: Percent bytes served to clients by all publishers, listed in decreasing order.

our methodology is based on the referer fields in the requests for CDN-delivered objects. I.e., there might be even more CDN-customer data being delivered than we are estimating.

We start with presenting some basic characteristics of the data sets from the two user populations covering all monitored subnets, see Figure 13. Overall we, in the UNI and EDU data sets, observed roughly 41 million different requests for Web objects for more than 330 Gbytes of data. This implies that the mean object size in our data sets is about 8 Kbytes. The mean size of an object served by the CDN to the users at UNI and EDU is somewhat larger at roughly 13.6 Kbytes. This explains that while only

1.7% respectively 2.75% of the requests are satisfied by the CDN, this accounts for 3.14% or 3.75% of the total bytes. The fact that users are accessing at least 1.4/2.0 times the amount of data directly from the publisher indicates that publishers do not delegate all of their traffic to the CDN. Therefore our approach for estimating publisher traffic can be expected to yield estimates of interesting interdomain traffic flows for a significant fraction of the overall traffic volume.

The large fraction of bytes in the category *related to non-CDN-customers* might, on first thought, be surprising. But actually there are two causes: publishers offload some of the content to other service providers, e.g., those providing targeted advertisement, and some of the publisher's content is served in connection with other sites, e.g., advertisements on someone else's Web page. While this indicates a lot of potential (up to 25% of all bytes, with regards to estimating traffic based on these traffic flows), we in this initial exploration phase focus on the ratio of CDN traffic on behalf of a publisher vs. the traffic to the publisher itself.

For this purpose we need to associate the bytes served by the CDN and the bytes related to CDN customers with the appropriate publisher. Using the appropriate information from the CDN this was accomplished without complications. Unfortunately while 41 million requests are quite a sizable number, the individual number of requests for objects served by the CDN per smaller time period (2 – 3 hours) are significantly reduced. Averaged over the whole duration of the trace collection there are only about 3.000 – 5.000 requests left for each data set and each two hour time period. Of course just averaging is unfair since there will be many more requests during busy hours than during off-hours, e.g., in the middle of the night. In addition some subnets, e.g., those with Web proxies, generated many more requests than others. Nevertheless it points out the problem of observing enough samples for deriving a reasonable ratio estimate.

Here we receive help from a trend that has been observed in many other contexts: We found the popularity of the Web content from some publishers to be much higher than the content from others. In Figure 14 (a), (b), we rank the number of requests and bytes by provider from the largest to smallest for both data sets, and plot the percentage of total requests/bytes attributed to each. For those publishers that contribute the most bytes these plots are nearly linear on a log-log scale. Again this is the characteristic of a Zipf-like distribution.

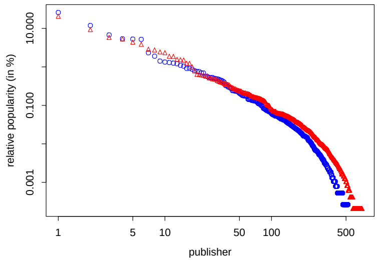
Together these two observations imply that we can expect to find time periods with a reasonable number of observations for some significant

Users	Description	Requests (in K)		Bytes (in Mbytes)	
		absolute	relative	absolute	relative
EDU	Total	27.807	100.00%	236.444	100.00%
UNI	Total	13.828	100.00%	105.833	100.00%
EDU	CDN	479	1.72%	7.431	3.14%
UNI	CDN	380	2.75%	3.973	3.75%
EDU	Related to CDN customer	1.190	4.28%	14.698	6.22%
UNI	CDN customer	603	4.36%	5.573	5.27%
EDU	Related to non-CDN-cust.	3.835	13.79%	59.941	25.35%
UNI	non-CDN-cust.	1.510	10.92%	19.260	18.20%

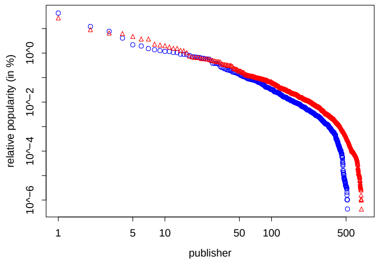
Figure 13: Basis statistics of the user access characteristics.

subset of the publishers in our user access data sets. For our further analysis we only consider those time periods with enough observations per publisher. Here we define “enough” as observing at least 100 requests satisfied by the CDN on behalf of a publisher and 100 request served by the publisher himself for an aggregation period. Using a value of 100 is rather arbitrary. Further analysis of these and other traces is needed to provide a sound basis for a good choice of this cutoff value. Nevertheless using our selection criteria we computed the ratios of bytes for each publisher and each aggregation period. Not too surprisingly we found that the ratios span quite a wide range of values: from 0.01 to 100. Since it is difficult to compare ratios (e.g., 0.06 vs. 0.03 corresponds to the same “difference” as 16 vs. 32) we, in all further discussion, use the binary logarithm of the ratios. Accordingly the values change: 0.06 to -4 vs. 0.03 to -5 and 16 to 4 vs. 32 to 5 and equal amounts correspond to zero. Figure 14 (c) plots the density of the ratios for the UNI data set for both bytes as well as requests. We observe for both data sets that the ratios span a significant range of values from -10 to 10 both for requests as well as for bytes. This indicates that different providers use different policies with regards to delegating their information to the CDN. Furthermore we see, as expected, that the CDN usually provides more bytes than the original publisher for most but not all publishers. In addition with regards to requests the distribution is much more balanced. This indicates that some publishers use the CDN for big objects, such as software distribution.

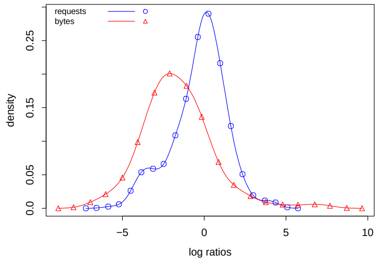
While the overall distribution of the ratios is interesting more relevant for the purpose of estimating the publisher demands is the question: How stable are the ratios across time and user populations? Overall it is well known that traffic volume [4] and flow arrival streams [73] are self-similar



(a) Requests



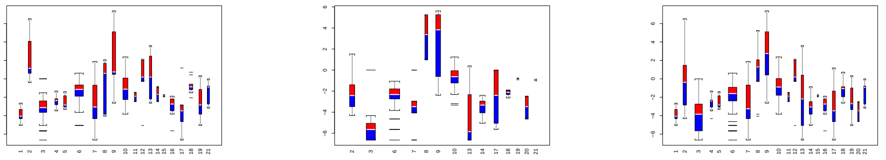
(b) Bytes



(c) Density

Figure 14: (a)/(b) Percent requests/bytes of user accesses to publishers, listed in decreasing order. (c) Density of the $\log_2$(ratios) reflecting the relationship between CDN and CDN customer traffic for UNI.

and exhibit quite some burstiness. Therefore we can expect some fluc-



UNI subset of EDU subset of EDU and UNI

Figure 15: Boxplot of the log(ratios,2) for some publishers.

tuations with regards to the number of requests over time. In addition not every user will access the same pages from the publisher, and different subsets of pages will lead to different ratios in terms of bytes from the publisher and the CDN. But what are the impacts of all these various reasons of instability? Our estimation methodology allows us to explore the size of these instabilities since it will yield multiple samples of estimated ratio values for various publishers. Figure 15 (a) and (b) show boxplots of the ratios for a subset of the publishers for the data sets for one subnet of EDU and all subnets of UNI. Boxplots can be used to display the location, the spread and the skewness of several data sets in one plot: The box shows the limits of the middle half, the line represents the median, and the extreme points (outliers) are displayed as well. We note that most of the boxes have a rather small spread (less than two). But others have quite a sizable spread, e.g., index 8 of UNI. This is partially due to a fairly small sample size and partially due to the variability of different content that is offered by that publisher. Further aggregation and combining the information from different user sets can sometimes be helpful. For example Figure 15 (c) shows the boxplots for the combined data sets. The estimations of the ratios for index 8 and most other have stabilized as indicated by the smaller range of the box. Only for a small subset do we observe problems, e.g., index 14.

Overall we can summarize that it is possible to estimate the ratio of the publisher demand serviced by the CDN vs. the one serviced by the publisher. But there are drawbacks to this approach: A large number of requests needs to be monitored in order to derive reliable estimations. The estimations can vary across time and some attention has to be paid towards different subject/interest areas by different user sets. Furthermore not all user sets will access sufficiently many objects from all publishers that are customers of the CDN. Therefore this approach should be combined with other approaches for estimating the ratios, e.g, static exploration of the Web site and information from the publisher itself.

5.3 Mapping of publisher demand to Web traffic demands

The next step is applying the methodology in Figure 8 to map the publisher demands to Web traffic demands. The open question is: how well does the proposed methodology of mapping each client set and each hostname to a single server IP address work? This is a two step process. First we need to identify the set of IP addresses for each hostname. Then we need to identify which subset of the IP addresses to choose for each client set.

If a hostname is hosted by the CDN itself or if the infrastructure is using DNS round robin by itself, the latter step is simple. In the first case we know which IP address serves the traffic and in the second case all returned IP addresses are used. Using the data described in Section 4 we observe that of the 8.042 hostnames, 731 (9.1%) are hosted by CDN itself, 339 (4.2%) are using some form of proximity-aware load balancing, while 6.972 (86.7%) are consistently returning the same set of IP addresses. Of these 6.972 hostnames, 6.526 (93.6%) are returning a single IP address while 446 (6.4%) are utilizing DNS round robin. Most of these (372) are using two IP addresses while 26 are using more than five IP addresses. Therefore we have solved the problem for 95.8% of the considered hostnames. Overall this indicates that a sizable fraction of the traffic that we capture will be interdomain traffic.

This leaves us the with 339 hostnames hosted on a distributed infrastructure. To better understand this infrastructure, we show, in Figure 16, a histogram of the number of IP addresses and the number of ASs. We observe that most hostnames (95.6%) are only mapped to a small number of IP addresses (< 20). Indeed more than 52% are using only two distinct IP addresses. 59% of these are in a single AS while 41% are located in different ASs. Overall we find that all IP addresses from 124 hostnames are located in a single AS. This means that from the view point of interdomain routing, we will not be able to distinguish these demands.

To explore how the infrastructure of the remaining 224 hostnames is embedded in the Internet we studied the minimal distances of the ASs of the IP addresses of the distributed infrastructure to every reachable AS in the BGP table. In order to compute the distances we consider the contractual relationships as derived from the routing tables. Each AS path may only cross a single peering/sibling edge, and may never follow a customer to provider edge once it has followed a provider to peer edge. Any edge unclassified by our heuristic is treated as a “sibling/peer” link. We observe that providers that use more servers and distribute them in various AS indeed gain some benefits. The mean distance and the standard deviation to all other ASs is somewhat smaller. On the other hand it appears that the choice of the AS has at least as much impact regarding the average BGP distance as just adding a presence in another AS.

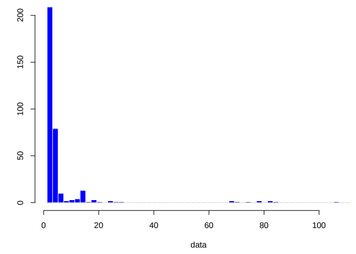
6 Summary

In this paper, we propose two models for interdomain traffic demands, publisher demands and Web traffic demands, that captures the volume of data, the origin as well as the destination of the data. Thus this simple abstraction can facilitate a wide range of engineering applications, ranging from traffic engineering, to planing of content delivery, to network simulation. We further present a methodology for populating parts of the demand model using logs from CDN networks, observations from user sets, the DNS and the routing system.

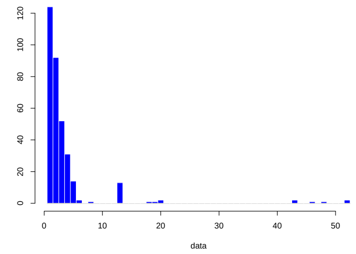
Our experimental results of applying the methodology to logs from a major CDN and two large user sets are promising. Our approach may allow us to capture a significant fraction of all Web traffic. But our initial results should be viewed mainly as an indication of the potential of the methodology – it is indeed possible to combine server log data from a CDN with packet level traces from large user sets to estimate a good chunk of all interdomain Web traffic. Indeed our results from mapping the publisher demands to Web demands indicates that our derived demands are indeed interdomain demands, show a large diversity and cover a sizable part of the Internet infrastructure.

At this point, however, specific numerical estimates should be viewed as tentative, as there is still work to be done. One question that has not yet been answered is whether the observed ratio of bytes served by the customer to bytes served by the CDN is reasonably invariant across diverse user sets. At this point we have examined only two. Another area in which our accuracy can be improved is in estimating which client blocks are served by which IP addresses for customers that employ sophisticated distributed infrastructures, e.g., using dynamic direction of clients to servers on the basis of external performance measurements.

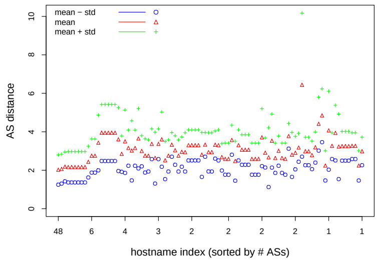
We have just started exploring the prospects of this kind of data, interdomain traffic demands. They can be used for such mundane analysis as time-of-day characterization, spatial distribution, analysis of user and



(a) IP addresses



(b) ASs



(c) AS distance to client sets

Figure 16: (a), (b) Histogram of number of different IP addresses/ASs per hostname. (c) AS distance to client sets for publishers with distributed infrastructures.

publisher dynamics, routing dynamics, etc. But we expect it to be even more fruitful to combine this data with other large data sets: In combination with BGP routing tables we can observe the paths that the individual traffic flows take through the Internet (at an AS level). With incorporating performance measurement data we can observe how flows change or BGP reacts (or does not react) in response to network bottlenecks.

Acknowledgments

We would like to thank numerous colleagues for helping us access and understand the network measurement and configuration data used in this work and for valuable suggestions on how to improve the presentation of the material.

References

[1] M. E. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: Evidence and possible causes," *IEEE/ACM Trans. Networking*, 1997.

[2] P. Barford, A. Bestavros, A. Bradley, and M. Crovella, "Changes in web client access patterns: Characteristics and caching implications," *World Wide Web, Special Issue on Characterization and Performance Evaluation*, 1999.

[3] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of Ethernet traffic," *IEEE/ACM Trans. Networking*, 1994.

[4] W. Willinger, V. Paxson, and M. S. Taqqu, "Self-similarity and Heavy Tails: Structural Modeling of Network Traffic," *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*, 1998.

[5] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang, "Experience in measuring backbone traffic variability: Models, metrics, measurements and meaning," in *Proc. ACM Measurement Workshop*, 2001.

[6] V. Paxson and S. Floyd, "Wide area traffic: The failure of poisson modeling," *IEEE/ACM Transactions on Networking*, 1995.

[7] M. Arlitt and C. Williamson, "Internet Web servers: Workload characterization and implications," *IEEE/ACM Trans. Networking*, 1997.

[8] A. Iyengar, M. S. Squillante, and L. Zhang, “Analysis and characterization of large-scale web server access patterns and performance,” *World Wide Web*, 1999.

[9] F. Douglis, A. Feldmann, B. Krishnamurthy, and J. Mogul, “Rate of change and other metrics: A live study of the World Wide Web,” in *Proc. USENIX Symp. on Internet Technologies and Systems*, 1997.

[10] C. E. Wills and M. Mikhailov, “Studying the impact of more complete server information on Web caching,” in *Proceedings of the 5th International Web Caching and Content Delivery Workshop*, 2000.

[11] D. O. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, “A framework for Internet traffic engineering.” Internet Draft draft-ietf-tewg-framework-01.txt, 2000.

[12] D. O. Awduche, “MPLS and traffic engineering in IP networks,” *IEEE Communication Magazine*, 1999.

[13] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, “Deriving Traffic Demands for Operational IP Networks: Methodology and Experience,” in *Proc. ACM SIGCOMM*, 2000.

[14] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, “Traffic Matrix Estimation: Existing Techniques and New Directions,” in *Proc. ACM SIGCOMM*, 2002.

[15] C. Barakat, P. Thiran, G. Iannaccone, and C. Diot, “On Internet backbone traffic modeling,” in *Proc. ACM SIGMETRICS*, 2002.

[16] A. Medina, C. Fraleigh, N. Taft, S. Bhattacharyya, and C. Diot, “A taxonomy of IP traffic matrices,” in *Workshop on Scalability and Traffic Control in IP Networks at the SPIE ITCOM+OPTICOMM Conference*, 2002.

[17] X. Xiao, A. Hannan, B. Bailey, and L. Ni, “Traffic engineering with MPLS in the Internet,” *IEEE Network Magazine*, 2000.

[18] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis, “Framework for IP performance metrics.” Request for Comments 2330, 1998.

[19] W. Stallings, *SNMP, SNMPv2, SNMPv3 and RMON 1 and 2.* Addison-Wesley, 1999.

[20] K. Thompson, G. J. Miller, and R. Wilder, “Wide-area internet traffic patterns and characteristics,” *IEEE Network Magazine*, 1997.

[21] D. Meyer and V. Gill, “Global routing operations bof,” 2003. `http://www.ietf.org/ietf/03mar/grow.txt`.

[22] T. E. Monk, “Inter-domain traffic engineering: Principles and case examples,” in *Proc. INET*, 2002.

[23] B. Fortz, J. Rexford, and M. Thorup, “Traffic engineering with traditional ip routing protocols,” in *IEEE Communication Magazine*, 2002.

[24] O. Maennel and A. Feldmann, “Realistic bgp traffic for test labs,” in *SIGCOMM*, 2002.

[25] C. Labovitz, “Scalability of the Internet backbone routing infrastructure,” in *PhD Thesis, University of Michigan*, 1999.

[26] T. G. Griffin and G. Wilfong, “An analysis of BGP convergence properties,” in *Proc. ACM SIGCOMM*, 1999.

[27] “Hot100.” `http://100hot.com/`.

[28] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, and J. Rexford, “NetScope: Traffic engineering for IP networks,” *IEEE Network Magazine*, 2000.

[29] S. Hull, *Content Delivery Networks: Web Switching for Security, Availability, and Speed.* McGraw-Hill, 2002.

[30] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl, “Globally distributed content delivery,” *IEEE Internet Computing*, 2002.

[31] S. Gadde, J. S. Chase, and M. Rabinovich, “Web caching and content distribution: a view from the interior,” *Computer Communications*, 2001.

[32] L. Bent and G. Voelker, “Whole page performance,” in *In Proceedings of the 7. International Workshop on Web Content Caching and Distribution*, 2002.

[33] K. L. Johnson, J. F. Carr, M. S. Day, and M. F. Kaashoek, "The measured performance of content distribution networks," in *Proceedings of the 5th International Web Caching and Content Delivery Workshop*, 2000.

[34] B. Krishnamurthy, C. Wills, and Y. Zhang, "On the use and performance of content distribution networks," in *Proc. ACM Measurement Workshop*, 2001.

[35] S. Saroiu, K. Gummadi, R. Dunn, S. Gribble, and H. Levy, "An analysis of internet content delivery systems," in *Proc. OSDI*, 2002.

[36] `http://www.akamai.com`.

[37] B. Halabi, *Internet Routing Architectures.* Cisco Press, 1997.

[38] J. W. Stewart, *BGP4: Inter-Domain Routing in the Internet.* Addison-Wesley, 1999.

[39] C. Huitema, *Routing in the Internet.* Prentice Hall, 1995.

[40] J. Moy, *OSPF: Anatomy of an Internet Routing Protocol.* Addison-Wesley, 1998.

[41] "Edge side includes." `http://www.esi.org/`.

[42] B. Lavoie and H. F. Nielsen, "Web characterization terminology & definitions sheet." `http://www.w3c.org/1999/05/WCA-terms/`.

[43] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski, "A flow-based model for Internet backbone traffic," in *Proc. ACM Measurement Workshop*, 2002.

[44] N. G. Duffield and M. Grossglauser, "Trajectory sampling for direct traffic observation," in *Proc. ACM SIGCOMM*, pp. 271–282, 2000.

[45] "A framework for passibe packet measurement." `http://www.research.att.com/projects/psamp/`.

[46] W. B. Teeuw, "The cdn state of the art.," 2001. `http://www.trc.nl/Middleware/cdn/ENindex.htm`.

[47] B. D. Davison, "Content delivery and distribution services," 2003. `http://www.web-caching.com/cdns.html`.

[48] `http://www.digitalisland.com`.

[49] `http://www.mirror-image.com`.

[50] L. Breslau, P. Cao, L. Fan, G. Philips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, 1999.

[51] W. Fang and L. Peterson, "Inter-AS traffic patterns and their implications," in *Proc. IEEE Global Internet*, 1999.

[52] "Zipf's law." `http://linkage.rockefeller.edu/wli/zipf/`.

[53] B. Krishnamurthy and J. Rexford, *Web Protocols and Practice*. Addison-Wesley, 2001.

[54] S. Manley and M. Seltzer, "Web facts and fantasy," in *Proc. USENIX Symp. on Internet Technologies and Systems*, 1997.

[55] E. Cohen, B. Krishnamurthy, and J. Rexford, "Improving end-to-end performance of the web using server volumes and proxy filters," in *SIGCOMM*, 1998.

[56] B. Krishnamurthy and C. Wills, "Study of piggyback cache validation for proxy caches in the WWW," in *Proc. USENIX Symp. on Internet Technologies and Systems*, 1998.

[57] V. N. Padmanabhan and J. C. Mogul, "Improving HTTP latency," *Computer Networks and ISDN Systems*, 1995.

[58] T. Kroeger, D. Long, and J. Mogul, "Exploring the bounds of Web latency reduction from caching and prefetching," in *Proc. USENIX Symp. on Internet Technologies and Systems*, 1997.

[59] A. Feldmann, "Blt: Bi-layer tracing of HTTP and TCP/IP," in *Proc. WWW-9*, 2000.

[60] H. Balakrishnan, V. Padmanabhan, S. Seshan, M. Stemm, and R. Katz, "TCP behavior of a busy Internet server: Analysis and improvements," in *Proc. IEEE INFOCOM*, 1998.

[61] S. Gribble and E. Brewer, "System design issues for Internet middleware services: Deductions from a large client trace," in *Proc. USENIX Symp. on Internet Technologies and Systems*, 1997.

[62] R. Wooster, S. Williams, and P. Brooks, "HTTPDUMP: a network HTTP packet snooper." 1996.

[63] G. Mallan and F. Jahanian, "An extensible probe architecture for network protocol performance measurement," in *Proc. ACM SIGCOMM*, 1999.

[64] B. Liu, "A different approach to content delivery." `http://www.isp-planet.com/news/2001/routescience.html`.

[65] R. Sommer and A. Feldmann, "Netflow: Information loss or win?," in *Proc. ACM Measurement Workshop*, 2002.

[66] Cisco Netflow. `http://www.cisco.com/warp/public/732/netflow/index.html`.

[67] V. Jacobson, C. Leres, and S. McCanne. `tcpdump`, `ftp://ftp.ee.lbl.gov`, `1989`.

[68] V. Paxon and R. Pang, "The HTTP analyser." `http://www.icir.org/vern/bro-manual/node51.html`.

[69] V. Paxson, "Bro: A system for detecting network intruders in real-time," in *Computer Networks*, 1999.

[70] University of Oregon RouteViews project. `http://www.routeviews.org/`.

[71] RIPE's Routing Information Service Raw Data Page. `http://data.ris.ripe.net/`.

[72] L. Gao, "On inferring autonomous system relationships in the Internet," in *Proc. IEEE Global Internet*, 2000.

[73] A. Feldmann, "Characteristics of TCP connection arrivals," in *Self-Similar Network Traffic And Performance Evaluation* (K. Park and W. Willinger, eds.), J. Wiley & Sons, Inc. 2000.