Verdiction of Time Delay in Wide-Area CDNs

C.Chandravathi
Assistant Professor
Department of IT
Vel Tech High Tech
Dr.Rangarajan Dr.Sakunthala
Engineering College

Fauzia Begam. K
Department of IT
Vel Tech High Tech
Dr.Rangarajan Dr.Sakunthala
Engineering College

Shamili. A
Department of IT
Vel Tech High Tech
Dr.Rangarajan Dr.Sakunthala
Engineering College

ABSTRACT

A content delivery network or content distribution network (CDN) is a server setup which allows for faster, more efficient delivery of your media files. It does this by maintaining copies of your media at different points of presence (POPs) along a global network to ensure quick client access and the fastest delivery possible. What you don't want is for your users to have to wait long periods of time while your photos or videos are downloaded.Minimizing user-perceived latency is crucial for Content Distribution Networks (CDNs) hosting interactive services. Latency may increase for many reasons, such as interdomain routing changes and the CDN's own load-balancing policies.

Keywords: Network diagnosis, latency increases, content distribution networks (CDNs).

1. INTRODUCTION

CDNs need greater visibility into the causes of latency increases, so they can adapt by directing traffic to different servers or paths. In this paper, we propose techniques for CDNs to diagnose large latency increases, based on passive measurements of performance, traffic, and routing.Separating the many causes from the effects is challenging. We propose a decision tree for classifying latency changes, and determine how to distinguish traffic shifts from increases in latency for existing servers, routers, and paths. Another challenge is that network operators group related clients to reduce measurement and control overhead, but the clients in a region may use multiple servers and paths during a measurement interval. We propose metrics that quantify the latency contributions across sets of servers and routers. Analyzing a month of data from Google's CDN, it is found that nearly 1% of the daily latency changes increase delay by more than 100 msec. More than 40% of these increases coincide with interdomain routing changes, and more than one-third involve a shift in traffic to different servers. This is the first work to diagnose latency problems in a large, operational CDN from purely passive measurements. Through case studies of individual events, we identify research challenges for measuring and managing wide-area latency for CDNs.

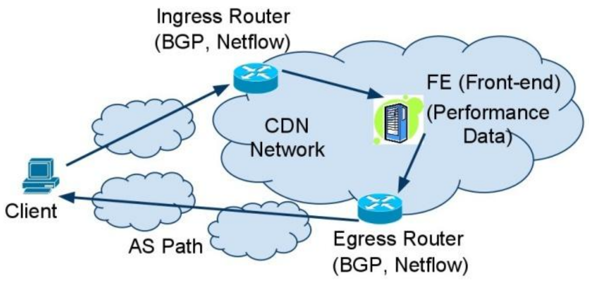

Fig. 1: CDN Architecture and measurements

To detect and diagnose latency problems, CDNs could deploy a large-scale active-monitoring infrastructure to collect performance measurements from synthetic clients all over the world. Instead, this paper explores how CDNs can diagnose latency problems based on measurements they can readily and efficiently collect passive measurements of performance, traffic, and routing from their own networks. Our goal is to design the system to maximize the information the CDN can glean from these sources of data. By joining data collected from different locations, the CDN can determine where a client request enters the CDN's network, which front-end server handles the request, and what egress router and interdomain path carry the response traffic, as shown in Fig. 1. Using this data, we analyze changes in wide-area latency between the clients and the front-end servers; the rest of the user-perceived latency, between the front and back-end servers, is already under the CDN's direct control. Finding the root cause of latency increases is difficult. Many factors can contribute to higher delays, including internal factors like how the CDN selects servers for the clients, and external factors such as interdomain routing changes. Moreover, separating cause from effect is a major challenge. For example, directing a client to a different front-end server naturally changes where traffic enters and leaves the network, but the routing system is not to blame for any resulting increase in latency. After detecting large increases in latency, our classification must first determine whether client requests shifted to different front-end servers, or the latency to reach the existing servers increased. Only then can we analyze why these changes happened.

2. DESIGN OF THE LATLONG TOOL

Based on the design, we implement a tool called LatLong, a tool for diagnosing large latency increases for CDN. We use LatLong to analyze a month of data from Google's CDN, and find that nearly 1% of the daily latency changes increase delay by more than 100 msec. Note that the latency increase of 100 msec is significant, since these are daily averages over groups of clients, and we only focus on latency-sensitive traffic for our study. More than 40% of these increases coincide with interdomain routing changes, and more than one-third involve a shift in traffic to different servers. This is the first work to diagnose latency problems in a large, operational CDN from purely passive measurements. Through case studies of individual events, we identify research challenges for managing wide-area latency for CDNs.

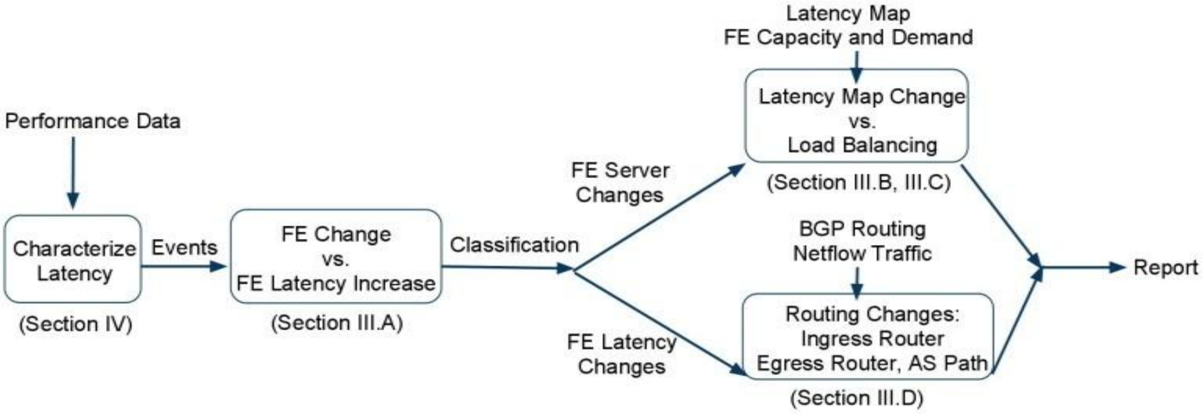

Fig. 2: Lat long system design

Content Distribution Networks (CDNs) offer user's access to a wide variety of services, running on geographically distributed servers. Many web services are delay- sensitive interactive applications (e.g., search, games, and collaborative editing). CDN administrators go to great lengths to minimize user-perceived latency, by over provisioning server resources, directing clients to nearby servers, and shifting traffic away from overloaded servers. Yet, CDNs are quite vulnerable to increases in the wide-area latency between their servers and the clients, due to interdomain routing changes or congestion in other domains. The CDN administrators need to detect and diagnose these large increases in round-trip time, and adapt to alleviate the problem (e.g., by directing clients to a different front-end server or adjusting routing policies to select a different path).To detect and diagnose latency problems, CDNs could deploy a large-scale active-monitoring infrastructure to collect performance measurements from synthetic clients all over the world. Instead, this paper explores how CDNs can diagnose latency problems based on measurements they can readily and efficiently collect—*passive* measurements of performance, traffic, and routing from their own networks. Our goal is to design the system to maximize the information the CDN can glean from these sources of data. By joining data collected from different locations, the CDN can determine where a client request enters the CDN's network, which front-end server handles the request, and what egress router and interdomain path carry the response traffic. Using this data, we analyze changes in wide-area latency between the clients and the front-end servers; the rest of the user-perceived latency, between the front and back-end servers, is already under the CDN's direct control. Finding the *root cause* of latency increases is difficult. Many factors can contribute to higher delays, including internal factors like how the CDN selects servers for the clients, and external factors such as inter-domain routing changes. Moreover, separating cause from effect is a major challenge. For example, directing a client to a different front-end server naturally changes where traffic enters and leaves the network, but the routing system is not to blame for any resulting increase in latency. After detecting large increases in latency, our classification must first determine whether client requests shifted to different front-end servers, or the latency to reach the existing servers increased. Only then can we analyze *why* these changes happened.

3. DISTRIBUTION OF LATENCY CHANGES

In the rest of the paper, we apply our tool to measurement data from Google's CDN. The BGP and Netflow data are collected and joined on a 15-minute timescale; the performance data is collected daily, and joined with the routing and traffic data to form a joint data set for each day in June 2010. For our analysis, we focus on the large latency increases which last for a long time and affect a large number of clients. We pick daily changes as the timescale, because the measurement data we get is aggregated daily. We group clients by "region," combining all IP addresses with the same origin AS *and* located in the same country. In this section, we describe how we preprocess the data, and characterize the distribution of daily increases in latency to identify the most significant events which last for days. We also determine the threshold for the large latency increases we study. As our datasets are proprietary, we are not able to reveal the exact number of regions or events, and instead report percentages in our tables and graphs; we believe percentages are more meaningful, since the exact number of events and regions naturally differ from one CDN to another. In addition, the granularity of the data, both spatially (i.e., by region) and temporally (i.e., by day) are beyond our control; these choices are not fundamental to our methodology, which could easily be applied to finer-grain measurement data. For a better understanding of large latency increases, we explore several events in greater detail.

These case studies illustrate the general challenges CDNs face in minimizing wide-area latency and point to directions for future work. Although many of these problems are known already, our case studies highlight that these issues arise in practice and are responsible for very large increases in latency affecting real users.

4. LAT LONG DIAGNOSIS OF LATENCY INCREASES

In this section, we apply our tool to study the events of large latency increases, which are identified in the previous section. We first classify them into FE changes and latency increases at individual FEs. Then, we further classify the events of FE changes according to the causes of the latency map and load balancing; classify the events of FE latency increases according to the causes of inter-domain routing changes. Our high-level results in this section are summarized in Table V. Nearly three-quarters of these events were explained (at least in part) by a large increase in latency to reach an existing front-end server. These latency increases often coincided with a change in the ingress router or egress router (or both!); still, many had no visible interdomain routing change and were presumably caused by BGP routing changes on the forward path or by congestion or intra-domain routing changes. Around one-third of the events involved a significant shift of client traffic to different front-end servers, often due to load-balancing decisions or changes in CDN's own view of the closest server. Nearly 9% of events involved both an "FE latency increase" *and* an "FE server change," which is why they sum to more than 100%.

5. CASE STUDIES

Future work on diagnosing wide-area latency increases for CDNs are explained below:

Direct extensions of our measurement study: First, we plan to extend our design to distinguish between routing changes that affect the egress router from those that only change the AS path. Second, as discussed at the end of Section V-C, we plan to further explore the unexplained shifts in traffic from one front-end server to another. Third, our case studies in Section VI required manual exploration, after automatically computing the various metrics. We plan to conduct more case studies and automate the analysis to generate reports for the network operators.

More accurate diagnosis: First, we plan to work with the groups that collect the measurement data to provide the data on a smaller timescale (to enable finer-grain analysis) and in real time (to enable real-time analysis). Second, we plan to explore better ways to track the performance data (including RTT and RPD) separately for each ingress router and egress/AS-path.

Currently, the choices of ingress and egress routers are not visible to the front-end servers, where the performance data are collected. Third, we will explore techniques for correlating across latency increases affecting multiple customer regions. CDNs have been widely deployed to serve Web content. In these systems, clients are directed to different servers to reduce latency and balance load. Our classification reveals the main causes of high latency between the clients and the servers. An early work in studied the effectiveness of DNS redirection and URL rewriting in improving client performance. This work characterizes the size and the number of the web objects CDNs served, the number of distinct IP addresses used in DNS redirection, and content download time, and compared the performance for a number of CDN networks.

For a better understanding of large latency increases, we explore several events in greater detail. These case studies illustrate the general challenges CDNs face in minimizing wide-area latency and point to directions for future work. Although many of these problems are known already, our case studies highlight that these issues arise in practice and are responsible for very large increases in latency affecting real users' events with a large increase in round-trip time. To identify the cause of latency increases, we first show the CDF of ΔFE (traffic shift) and ΔLat (latency increase) for the events we study in Figure 4. The distributions are a reflection of each other (on both the x and y axes), because ΔFE and ΔLat sum to 1 for each event. The graph shows that about half of the events have ΔFE below 0.1, implying that shifts in traffic from one FE to another are not the major cause of large-latency events. Still, traffic shifts are responsible for *some* of the latency increases—one event has a ΔFE of 5.83! (Note that we do not show the very few points with extreme ΔFE or ΔLat values, so we can illustrate the majority of the distribution more clearly in the graph). In comparison, ΔLat is often fairly high—in fact, more than 70% of these events have a ΔLat higher than 0.5. To classify these events, we apply a threshold to both distributions and identify whether ΔFE or ΔLat (or both) exceeds the threshold. Table V-A summarizes the results for thresholds 0.3, 0.4, and 0.5. These results show that for a range of thresholds, around two-thirds of the events are explained primarily by an increase in latency between the clients and the FEs. For example, using a threshold of 0.4 for both distributions, 65% of events have a large ΔLat and another 9% of events have large values for both metrics, resulting in nearly three-quarters of the events caused (in large part) by increases in RTTs to select front-end servers.

5.1 Normal Front-End Changes

To understand the normal distribution of latency-map changes, we calculate $\Delta LatMap$ for *all* of the regions— whether or not they experience a large increase in latency— on two consecutive days in June 2010. Figure 5 shows the results. For 76.9% of the regions, less than 10% of the requests change FEs because of changes to the latency map. For 85.7% of regions, less than 30% of traffic shifts to different front- end servers. Less than 10% of the regions see more than half of the requests changing front-end servers. Often, these changes involve shifts to another front-end server in a nearby geographic region.

5.2 Latency-Map Inaccuracies

During one day in June 2010, an ISP in the United States saw the average round-trip time increase by 111 msec. Our analysis shows that the RTT increased because of a shift of traffic to different front-end servers; in particular, ΔFW *was* 1.01. These shifts were triggered primarily by a change in the latency map; in particular, $\Delta LatMap$ was 0.90. Looking at the latency map in more detail revealed the reason for the change. On the first day, 78% of client requests were directed to front-end servers in the United States, and 22%

were directed to servers in Europe. In contrast, on the second day, all requests were directed to front-end servers in Europe. Hence, the average latency increased because the clients were directed to servers that were further away. The situation was temporary, and the clients were soon directed to closer front-end servers. Clients do not necessarily reside near their local DNS servers, especially with the increasing use of services like GoogleDNS and OpenDNS. Similarly, client IP addresses do not necessarily fall in the same IP prefix as their local DNS server. Further, DNS caching causes the local DNS server to return the same IP address to many clients over a period of time. All of these limitations of DNS make it difficult for a CDN to exert fine-grain control over server selection. Recent work at the IETF proposes extensions to DNS so requests from local DNS servers include the client's IP address [10], which should go a long way toward addressing this problem. Still, further research on efficient measurement techniques and efficient, fine-grain control over server selection would be very useful.

6. FUTURE ENHANCEMENT

In this section, we briefly discuss several natural directions for future work on diagnosing wide-area latency increases for CDNs.

Direct extensions of our measurement study: First, we plan to extend our design in Section III to distinguish between routing changes that affect the egress router from those that only change the AS path. Second, as discussed at the end of Section V-C, we plan to further explore the unexplained shifts in traffic from one front-end server to another. We suspect that some of these shifts are caused by a relatively small fraction of traffic shifting to a much further away front-end server. To analyze this further, we plan to incorporate the RTT differences between front-end servers as part of our metrics for studying FE changes. Third, our case studies in Section VI required manual exploration, after automatically computing the various metrics. We plan to conduct more case studies and automate the analysis to generate reports for the network operators.

More accurate diagnosis: First, we plan to work with the groups that collect the measurement data to provide the data on a smaller timescale (to enable finer-grain analysis) and in real time (to enable real-time analysis). Second, we plan to explore better ways to track the performance data (including RTT and RPD) separately for each ingress router and egress/AS-path. Currently, the choice of ingress and egress routers is not visible to the front-end servers, where the performance data are collected. Third, we will explore techniques for correlating across latency increases affecting multiple customer regions. For example, correlating across inter-domain routing changes that affect the AS paths for multiple client prefixes may enable us to better identify the root cause.

Incorporating additional data sets: We plan to investigate techniques for improving the visibility of the routing and performance changes from outside the CDN network. For example, active measurements such as performance probes and trace-route would help explain the "unknown" category for the ΔLat events, which we could not correlate with visible routing changes. In addition, measurements from the front- end servers could help estimate the performance of alternate paths, to drive changes to the CDN's routing decisions to avoid inter-domain paths offering poor performance.

7. CONCLUSION

The Internet is increasingly a platform for users to access online services hosted on servers distributed throughout the world. Today, ensuring good user-perceived performance is a challenging task for the operators of large Content Distribution Networks (CDNs). In this paper, we presented the system design for automatically classifying large changes in wide-area latency for CDNs, and the results from applying our methodology to traffic, routing, and performance data from Google. Our techniques enable network operators to learn quickly about significant changes in user-perceived performance for accessing their services, and adjust their routing and server-selection policies to alleviate the problem.

Using only measurement data readily available to the CDN, we can automatically trace latency changes to shifts in traffic to different front-end servers (due to load-balancing policies or changes in the CDN's own view of the closest server) and changes in the inter-domain paths (to and from the clients). Our analysis and case studies suggest exciting avenues for future research to make the Internet a better platform for accessing and managing online services.

8. REFERENCES

[1] M. Szymaniak, D. Presotto, G. Pierre, and M. V. Steen, "Practical large- scale latency estimation", *Computer Networks*, 2008.

[2] R. Krishnan, H. V. Madhyastha, S. Srinivasan, and S. Jain, "Moving beyond end-to-end path information to optimize CDN performance," in *Proc. 2009 Internet Measurement Conference*.

[3] Z. M. Mao, C. Cranor, F. Douglis, M. Rabinovich, [3] O. Spatscheck, and J. Wang, "A precise and efficient evaluation of the proximity between web clients and their local DNS servers," in *Proc. 2002 USENIX Annual Technical Conference*.

[4] E. Katz-Bassett, H. V. Madhyastha, V. K. Adhikari, C. Scott, J. Sherry, P. van Wesep, T. Anderson, and A. Krishnamurthy, "Reverse Traceroute," in *Proc. 2010 Networked SystemsDesign and Implementation*.

[5] B. Krishnamurthy, C. Wills, and Y. Zhang, "On The use and performance of content distribution Networks," in *Proc. 2001 Internet Measurement.*