Research Statement

Balakrishnan Chandrasekaran
(balac@mpi-inf.mpg.de)

The majority of the Internet traffic today is delivered via content delivery networks (CDNs). The ever increasing end-user demands for bandwidth and latency seem to assure that the volume of content delivered through CDNs will only increase going forward. Typically, the CDN hauls data from origins (i.e., content providers) to its *back-end* servers, moves this data (over a sophisticated overlay network) to its *front-end* servers, and serves the data from there to the end users. If we focus on the path taken by content from origins to end users via a CDN a simple fact becomes apparent: a significant fraction of this path traverses the CDN infrastructure, between the back-end and front-end servers. The server-to-server "landscape" formed by these paths is increasingly becoming "longer," as front-end servers are being deployed closer to end users to minimize the *last mile* latency. Since CDNs have been expanding their infrastructure by deploying more servers in diverse networks and geographic locations, the landscape is also becoming "wider." This server-to-server landscape provides a few unique perspectives as well as opportunities for solving some long-standing networking problems in the Internet. My research revolves around characterizing this server-to-server landscape, leveraging the unique perspectives offered by this landscape to provide insights into the Internet's structure and performance of protocols deployed on the Internet, and, lastly, exploiting the landscape to improve the Internet's performance and security. I will begin by discussing my current research efforts, highlighting in each a few open questions to hint at the short-term goals, and conclude with an outlook on the long-term research agenda.

Characterizing the Server-to-Server Landscape

My earliest work on the server-to-server landscape was on characterizing the Web traffic exchanged between servers. I collaborated with researchers from Akamai Technologies (one of the largest CDNs in the World) and Technische Universität Berlin to estimate and characterize the *back-office Web traffic* [6], which refers to the Web traffic between front-end servers (from where data is directly delivered to the end users) and any other server in the back end that is be part of the Web ecosystem. Our work used measurements from a diverse set of vantage points, including IXPs, ISPs, access networks, and CDNs, and revealed that delivering content to the end users involves a rich and complex interaction between many servers. Advertisement objects, for instance, result from an online bidding process involving several servers before these objects are delivered to the end users. Web objects served by CDNs similarly may involve interactions across a hierarchy of servers before being delivered to end users. Such server-to-server interactions, our study showed, represent a significant fraction of the core Internet traffic.

The data set from CDN consisted of Web server logs from servers at different geographically diverse locations. While our study combined data from multiple vantage points, the CDN's perspective alone revealed several key insights. One such insight was that while a significant fraction of the traffic (i.e., at least 25%) observed at each server cluster constituted back-office Web traffic, the percentage of back-office traffic varies significantly across the clusters; it is non-trivial to reason about these variations without knowledge of how the content is delivered by the CDN. Web-server logs from servers in Frankfurt, for instance, indicated that back-office traffic accounted for more than 70% of the overall traffic, while servers from nearby Paris showed only roughly 20% back-office traffic. Even more surprisingly, only 12% of the overall traffic at Frankfurt was between the CDN's (front-end) servers and end users. One naïve interpretation of these observations might be that the caching at the Frankfurt was inefficient. We provided a more accurate explanation: Frankfurt was caching content for another server cluster—in this case, Paris; rather than having both the server clusters—one at Frankfurt and the other at Paris—fetch traffic over transatlantic links, the CDN was configured to incur the cost only once—at Frankfurt. The data set also shows that a non-trivial fraction of the traffic between servers is invisible to the public Internet (e.g., intra-cluster or inter-rack traffic).

The CDN's perspective, as seen in above example, has several implications for network-measurement efforts. Server-to-server traffic constitutes a significant fraction of the Internet's traffic and it is hard to characterize this traffic solely from measurements gathered at IXPs and ISPs. A deeper understanding of the concerned CDNs and their operation is crucial for providing accurate explanations of the observations. The data set also shows that Web objects may traverse a hierarchy of CDN servers (e.g., from servers in Frankfurt to those in Paris, in the above example, before being delivered to the end users near Paris); these configurations (i.e, which server cluster caches for which and when) may also change over time depending on various factors. These observations highlight the need for a more systematic approach for measuring and characterizing the performance of CDNs. Unsurprisingly, measuring the server-to-server traffic over time, globally across the entire Internet, even for one CDN is challenging. Generating a traffic matrix that succinctly captures the dynamics of server-to-server traffic will, however, be immensely useful for the networking community; it could, for instance, inform the design and evaluation of network protocols for the server-to-server landscape.

Leveraging the Server-to-Server View

Since the delivery of content involves significant server-to-server communications, it was only natural to followup the previous work with an effort to measure the performance of the (server-to-server) paths traversed by the content. Additionally, by carefully choosing the servers from a CDN infrastructure, to represent a diverse mix from servers from different networks and geographies, we can use the server-to-server paths as a proxy to measure the performance of the Internet's core (i.e., end-to-end latency of paths across the Internet's core). Indeed, in collaboration with researchers from Akamai, MIT, and CAIDA, I gathered nearly 1.2 billion (traceroute and ping) measurements over both IPv4 and IPv6 protocols between 646 CDN servers, located in diverse geographic locations and networks, across a period of 16 months. We analyzed the impact of routing changes and congestion (in the core) on the end-to-end (i.e., server-to-server path) latencies [4]. The study highlighted that, for the most part, server-to-server paths experienced only a few AS-level changes (30 or fewer, across the study period), and these changes typically did not significantly increase the end-to-end round-trip times. Given that we selected only dual-stack servers from the CDN's infrastructure, we were also able to show how the performance between the same pair of servers differed across the protocols. Summarizing at a high level, our empirical observations revealed that in most cases the performance did not differ much; we identified a small percent of paths where switching from one protocol to the other will yield better performance (or lower end-to-end latency). Whether one protocol dominates the other (by providing a comparatively lower end-to-end latency) consistently over time, or if these performance variations are transient is yet unknown. We also have not investigated to what extent infrastructure sharing between the paths over the two protocols explain the discrepancies; stated differently, to what extent do the paths over IPv4 and IPv6, between any pair of servers, share the same infrastructure (i.e., routers) still remains unknown.

Any network congestion in the Internet's core (for which the server-to-server paths serve as a proxy) will increase the end-to-end latencies and adversely impact the end-user experiences. Further, congestion in the core is a hotly debated topic—some claim it to be the norm, some argue that it is only prevalent at peak times, and some completely dismiss the notion of a congested core. To gain insight, we looked for *consistent congestion*, defined as recurring daily oscillations in round-trip times, in our data set and showed that it is *not* the norm in the core. Between congestion and (AS-level) routing changes, our empirical study suggested that, between a pair of servers, the latter typically increase the end-to-end latencies more compared to the former. Although our (consistent) congestion inferences are based on well-studied, widely used delay-based methods, the caveats of such delay-based inferences are also well-known. With major content providers and CDNs pushing for a more widespread adoption and use of explicit congestion notification (ECN) signals, it will be worthwhile to revisit the study of congestion, but with ECN-based inferences, in the core. An even more challenging undertaking will be to combine the perspectives from multiple vantage points, including IXPs, ISPs, and CDNs, to estimate an Internet-wide congestion heat-map. Such a heat-map will inform policies regarding how networks peer and how we perform traffic engineering in the different networks, and the "hot spots" will highlight areas that need further optimization.

Exploiting new opportunities

The server-to-server landscape presents a novel test-bed for designing solutions to some of the long-standing networking problems in the Internet. These opportunities stem from a simple fact: the server-to-server landscape, even when spread across several autonomous systems or networks, is still logically one entity, within the control of a singe organization. In other words, since the endpoints (or servers) in this landscape are within the control of a single organization (i.e., the CDNs that owns the servers), it is possible to design, experiment, and validate network protocols that would otherwise be infeasible to be deployed, even for testing, in the Internet. Unlike other test-beds, the server-to-server landscape only offers control of the endpoints; the paths between the endpoints might still be subject to the vagaries of the larger Internet. Additionally, suppose CDNs control the software on end-user machines (e.g., a JavaScript-based application that fetches Web objects), we can also revisit some of the sub-optimal protocol choices in delivering content to end users that have so far remained ossified.

Indeed, as part of an ongoing effort at the Max-Planck-Institut für Informatik, I'm leveraging the server-to-server landscape to question the status quo in video streaming. Today, TCP is the dominant transport protocol for video streaming, due to the widespread use of dynamic adaptive streaming over HTTP (DASH). The rich body of prior work on optimizing TCP, adaptive bitrate selection algorithms, or TCP variants, however, highlights TCP's shortcomings. Our preliminary investigation reveals that even at a loss rate of 0.16%—lower than that typically observed in the Internet—the video player spends 20% of the total video time in stalls (i.e., in waiting for the lost packets to arrive at the playback buffer) when streaming using TCP [5]. But with CDNs (e.g., Akamai Technologies) and popular Web browsers (e.g., Google Chrome) already supporting QUIC, it is worth revisiting this status quo in streaming.

A simple observation should, nevertheless, highlight that reliable transports are ill-suited for video streaming: video data consists of different types of frames, some types of which do *not* require reliable delivery. The loss of some types of frames has minimal or no impact (since such losses can be recovered) on the end-user quality of experience (QoE). Therefore, by adding support for *unreliable* streams in QUIC and offering a selectively reliable transport, wherein not all video frames are delivered reliably, we can optimize video streaming and improve end-user experiences. This approach has several advantages: (a) it builds atop QUIC that is rapidly gaining adoption; and (b) it involves only a simple, backward compatible, incrementally deployable extension—support for unreliable streams in QUIC [5]. I am currently working on a thorough examination of the use of unreliable streams for video streaming and the interplay between such a partially reliable transport with the application-layer adaptive bitrate schemes is still in the pipeline. For the latter, I envision my collaborations with experts in adaptive bitrate schemes at the University of Massachusetts, Amherst will be immensely helpful.

There are several other problems, besides video streaming, that could benefit from a design based on the server-to-server landscape. For instance, the idea of mapping the users to the "closest" server, (say, one that provides the lowest latency) has typically been using DNS tricks or anycast. A hybrid scheme combining the two would likely perform much better than either one, and the evaluation of such a technique—where it works or fails, and why—is well within the scope of my research agenda. Performance problems aside, numerous security issues could at least be addressed in part (since Internet-scale security solutions often need the coordinated efforts from multiple entities) and motivate other players to contribute towards improving security.

Future Directions

The server-to-server landscape offers tremendous potential for addressing some of the grand challenges of networking. We could use the massively distributed server infrastructure of a CDN, for example, to gather Internet-wide measurements, to follow a measurement-driven approach for designing truly scalable solutions, and even to test the effectiveness of solutions (to a large extent) by exploiting the logically monolithic, but highly distributed CDN infrastructure. To this end, my long term agenda is to make the Internet faster and more secure; my goal, more specifically, is to identify and evaluate the role of the server-to-server landscape in achieving these two closely related[1] goals. In the remainder of this section, I highlight a few initial explorations in this space and sketch the outlines of a few projects.

[1] Since latency could be one reason, as in the case of OCSP, for lack of adoption of a security protocol.

Towards a Speed-of-Light Internet

Reducing latency across the Internet is of immense value and content providers have clearly demonstrated the impact latency has on their bottom lines. Microsoft's Bing, for instance, found that a two second slowdown translated to a drop in revenue per user of 4.3%. Latency is also crucial to facilitate the idea of running software in the cloud. A low-latency network allows more computations to be offloaded to the cloud, while giving the end users an illusion that they are running their computations locally (on their own machines). The Internet, however, is shockingly slow today! I collaborated with researchers at Duke University, University of Illinois Urbana-Champaign, Yale University, University of California Santa Cruz, and ETH Zürich to quantify the latency "inflation" in the Internet, defined as the ratio of the observed round-trip time to the theoretical limit—the speed of light in free space [3]. The rich and diverse data sets we used in this work and shared with the community was awarded the "Best Dataset Award" at the 2017 Passive and Active Measurements (PAM) conference held in Sydney, Australia. Our study showed that the median time to fetch just the HTML documents of popular Web sites was 37-times slower than the round-trip speed-of-light latency (between the corresponding clients and servers) [3]. We showed that infrastructural improvements alone could reduce latency by at least a factor of three, which was further supported by independent observations from the earlier server-to-server study [4].

To show that it is economically feasible to build a speed-of-light (or *cSpeed*) network and demonstrate its utility, we outlined the design of a low-latency cSpeed Internet that connected the top 120 populous cities in United States using microwave communications [7]. We are currently investigating the design of such supplemental cSpeed networks (operating in parallel with the existing mostly fiber-based Internet) in both US and Europe. In particular, we are evaluating the cSpeed network's performance under adverse weather conditions and when subject to different traffic matrices [2]. While attempting to measurement the effect of reducing latency for applications such as Web browsing, we found that speeding up the client-server path alone could improve the end-user's browsing experiences (estimated by measuring the page-load times) [2]. It is only natural, hence, to consider what a CDN could do. A CDN is one potential "killer" application or use case for a cSpeed network: the objective of a CDN to deliver content as quickly as possible combined with their ability to identify the latency-sensitive content among the overall traffic makes CDNs an ideal user of the cSpeed network. How would the design of a CDN change, if it had access to a cSpeed (but bandwidth-limited compared to fiber) backbone, supplementing the CDN's existing connectivity? The earlier server-to-server work highlighted a rich interaction between servers involved in delivering content. We do not yet know how these interactions affect the delivery time of content. What fraction of the overall traffic carried by a CDN is latency sensitive, and how does this latency-sensitive traffic vary over time?

Plugging the Internet's security holes

The unprecedented growth of Internet-of-Things (IoT) devices and the emergence of a connected *Internet of Everything*, coupled with the fragility of the IoT ecosystem, provides a ripe platform for inimical parties to exploit the weaknesses and launch massive DDoS attacks. In this regard, the recent Mirai botnet attacks are only a harbinger of more widespread and crippling attacks in the future. Although several point solutions have been proposed, they offer, if any, only a veneer of security, and we are in a dire need of a more distributed and concerted approach to secure the Internet [1]. DDoS attacks are also becoming increasingly sophisticated, with some "pulse wave" attacks intelligently regulating attack-traffic's volume to evade detection. Could we exploit the network and geographic diversity of a highly distributed CDN infrastructure to detect botnet-based DDoS attacks well before the attack-traffic volume ramps up? Is it feasible to run DDoS detection algorithms in a distributed fashion, rather than requiring traffic to be funneled to a centralized location for analyses? Today, DDoS mitigation solutions rely on anycast to ingest traffic from end users to the "nearest" datacenter where it is "scrubbed" to allow only legitimate traffic to proceed forward towards its intended destination. This "scrubbing" centers may, however, turn into "choke points" when attack volumes increase beyond a few hundred Gigabits per second. Although deploying more datacenters will delay the onset of such issues, we have not explored other alternatives. One potential approach for a CDN might be to coordinate, in the detection and mitigation of DDoS attacks, with an ISP or access network and drop (or blackhole) the malicious traffic close to the source. How can we facilitate such a coordination between a CDN and an ISP? Are BGP community attributes a good vehicle for exchanging traffic insights between a CDN and an ISP? Designing, testing, and evaluating the effectiveness of such solutions at scale will likely provide crucial insights for designing a more secure Internet than exists today.

References

[1] T. Benson and B. Chandrasekaran. Sounding the Bell for Improving Internet (of Things) Security. In *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*, IoTS&#38;P ’17, pages 77–82, New York, NY, USA, 2017. ACM.

[2] D. Bhattacherjee, S. A. Jyothi, I. N. Bozkurt, M. Tirmazi, W. Aqeel, A. Aguirre, B. Chandrasekaran, P. B. Godfrey, G. Laughlin, B. M. Maggs, and A. Singla. cISP: A Speed-of-Light Internet Service Provider. *CoRR*, abs/1809.10897, 2018.

[3] I. N. Bozkurt, A. Aguirre, B. Chandrasekaran, P. B. Godfrey, G. Laughlin, B. Maggs, and A. Singla. Why is the internet so slow?! In M. A. Kaafar, S. Uhlig, and J. Amann, editors, *Passive and Active Measurement: 18th International Conference, PAM 2017, Sydney, NSW, Australia, March 30-31, 2017, Proceedings*, pages 173–187, Cham, 2017. Springer International Publishing.

[4] B. Chandrasekaran, G. Smaragdakis, A. Berger, M. Luckie, and K.-C. Ng. A Server-to-Server View of the Internet. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, CoNEXT ’15, pages 40:1–40:13, New York, NY, USA, December 2015. ACM.

[5] M. Palmer, T. Krüger, B. Chandrasekaran, and A. Feldmann. The QUIC Fix for Optimal Video Streaming. In *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC*, EPIQ’18, pages 43–49, New York, NY, USA, 2018. ACM.

[6] E. Pujol, P. Richter, B. Chandrasekaran, G. Smaragdakis, A. Feldmann, B. M. Maggs, and K.-C. Ng. Back-Office Web Traffic on The Internet. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, IMC ’14, pages 257–270, New York, NY, USA, 2014. ACM.

[7] A. Singla, B. Chandrasekaran, P. B. Godfrey, and B. Maggs. The Internet at the Speed of Light. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, HotNets-XIII, pages 1:1–1:7, New York, NY, USA, 2014. ACM.