A Practical System for Revealing ISP Traffic Differentiation

David Choffnes[‡], Phillipa Gill[⋆], Alan Mislove[‡]
[‡]Northeastern University, [⋆]Stony Brook University

Abstract. Traffic differentiation empowers network operators to give different levels of service to different types of network flows, and can be used to manage QoE for end users. Without regulation or accountability, such network management practices could be used to raise the barrier to entry for new technologies, or block them entirely. As a result, the FCC recently passed new rules making differentiation illegal in broadband networks [2]. Further, by breaking end-to-end system design, these practices can have negative side-effects on reachability, reliability, and performance.

We are currently implementing a systematic approach to understand and expose traffic differentiation and its impact on the Internet, with the following main contributions. First, we have studied real traffic differentiation devices to understand the range of their behaviors, and have developed a testbed to test the effectiveness of our measurement techniques in a controlled environment. Second, we have designed and deployed systems to detect differentiation efficiently, to operate in both resource-constrained mobile environments and fixed-line settings. With a large-scale deployment that includes incentives for adoption by users and content providers, we will characterize the impact of traffic differentiation on network performance. Last, we are developing a model to understand how to detect traffic shaping implementations in general, and analyze large-scale impacts of differentiation.

Key challenges for differentiation detection. Our work addresses three key challenges that limit our understanding of traffic differentiation. First, researchers often struggle to characterize traffic differentiation because the products that implement it are generally closed systems, operators rarely acknowledge the presence of middleboxes in networks, and researchers have limited or no access to experiment with such middleboxes. Second, identifying and understanding differentiation behavior often requires measurements from vantage points inside the network (*e.g.,* from end users and on mobile devices), which limits the coverage and feasibility of any study that uses a fixed testbed (*e.g.,* PlanetLab or lab environments) and necessitates the development of easy-to-deploy systems to detect traffic differentiation. Finally, we lack models of how traffic differentiating products operate on network traffic, making it difficult to reason about interactions between multiple devices on a path.

A flexible, realistic testbed for experimenting with differentiation detection. We have studied devices that implement differentiation to understand their behavior, and developed a testbed that allows us to validate our measurement techniques in a controlled environment. A key challenge for detecting differentiation is how to support controlled experiments on network traffic from representative applications. Prior work focused on specific applications (*e.g.,* BitTorrent [1]) and manually generated traffic to match the target application. This approach, however, does not scale to large numbers of applications and may even be infeasible when the target application uses proprietary, or otherwise closed, protocols.

To address this, we use a *trace record and replay* approach [4]. We begin by recording a packet capture of the application. The packet capture is then parsed to extract the *application layer* messages exchanged. Importantly, we capture the application-level client–server interaction, which *does not* rely on properties of the underlying network (*e.g.,* packet loss, delay) used for recording. The corresponding messages are then replayed between a client and server running our replay software. Importantly, because our approach reproduces payloads from the *application layer*, it supports both encrypted (e.g., HTTPS, XMPP) and plaintext traffic. For example, our approach precisely replicates the encrypted stream of data recorded from Netflix traffic, without needing to know the corresponding plaintext. We have implemented this approach for six popular apps, and are developing support to create replay traces for arbitrary user-submitted traffic.

We have leveraged two commercial packet-shaping devices that we have acquired to conduct grey-box testing of differentiation, and we are working to acquire others. To broaden the utility of our testbed, we will

develop techniques to emulate shaping features in a scalable, open platform. An important step in validating our trace record and replay system is to understand how traffic differentiation devices classify traffic and whether our replays are able to capture salient features of the traffic to trigger differentiation.

We have used our initial testbed of differentiation devices to test whether we replay network traffic with sufficient fidelity to trigger shaping, and to test specifically *which features* of the replayed traffic cause differentiation. We designed a battery of tests using our trace replay framework to understand the features most relevant for traffic shaping. These tests include truncating the trace, prepending or appending the trace, inserting/modifying traffic within the trace, and spoofing destination IPs. As an example of the results, our analysis indicates that a commercial shaper in our testbed uses a simple regular expression match on the string "youtube" to classify YouTube traffic, and fails to classify this traffic if the request does not start with "GET" or the string does not appear in the first packet of the request.

Armed with information about which features of network traffic trigger shaping, we are investigating how to design more efficient measurement techniques to detect traffic differentiation outside of the testbed. Further, we can exploit this information to develop simple evasion tactics.

Efficient methods to measure traffic differentiation in a wide range of network environments. We posit that differentiation most likely occurs at eyeball networks. First, eyeball networks generate more inbound Internet traffic than outbound and tend to be stub networks, meaning traffic crossing their network results in charges from their providers. As a result, these networks have incentives to control traffic volumes traversing their border. Further, cellular data networks have constraints (*e.g.,* limited spectrum) that makes bandwidth management via differentiation appealing. Second, transit networks make money based on traffic volumes, so they have little incentive to interfere with traffic in any way that reduces volumes.

Thus, measuring differentiation requires access to vantage points in eyeball networks, *i.e.,* we need tools that run on user devices and machines. We propose using crowdsourcing to acquire these vantage points. To do so, we need client software that is easy to install, does not need root permissions, and is sensitive to resource limitations such as data caps and limited processing power. For a desktop client, we will explore using portable platforms such as Java, WebRTC, and Flash. However, these frameworks will not work in the mobile environment.

To address this limitation, we have built a mobile app called *Differentiation Detector*[1] that identifies differentiation as follows. First, it tests for differentiation by replaying real network traces from mobile clients (using the mechanism described above). Meddle [5] facilitates this, and we use this to replay arbitrary app traces. Second, *Differentiation Detector* exploits the Meddle VPN to conduct controlled experiments. By alternately replaying traffic over tunneled and untunneled connections multiple times in rapid succession, we control ISP visibility into packet contents that may be used to differentiate traffic.

A key challenge is how to establish ground truth as to whether the ISP is shaping the replayed traffic. To address this, we exploit the VPN connection that Meddle provides as follows. When the VPN is enabled, the ISP cannot inspect flow contents and thus cannot differentiate except based on total bandwidth and time-of-day (and, of course, VPN traffic). We then compare this performance to the case when we send traffic untunneled. Using multiple successive trials of tunneled and untunneled replay experiments, we can determine the noise inherent in performance metrics in each type of experiment (tunneled vs not tunneled), then identify cases where there are statistically significant differences between them, indicating differentiation.

We are using our approach to investigate several other factors related to differentiation, such as determining what applications are impacted, by how much, whether there are differences over time and in different regions of the same ISP, and how differentiation relates to users' data caps. After gathering measurements from different users in different locations over time, we will stratify the dataset to determine which key factors influence differentiation.

[1] `https://play.google.com/store/apps/details?id=com.stonybrook.replay`

Towards a general model of how differentiation occurs. As part of our ongoing work, we aim to create a taxonomy of how devices operate (*e.g.,* in-path, on-path), how they can classify traffic (*e.g.,* header-space analysis, payload-based identification) and shape it (*e.g.,* leaky bucket, RED, or packet injection). We will use this to design detection methods that are robust to different implementations of shaping. We will also examine how multiple devices using different techniques on the same network path may interact in unanticipated ways.

For example, Terzis [3] pointed out that T-Mobile uses a Web proxy that sends traffic to a user on a path that includes a shaper, but the proxy was sending data faster than the peak rate allowed by the shaper, leading to significant packet loss and retransmissions. We will use our model of shaper behaviors and TCP behavior to understand pathological cases when multiple middleboxes appear on the same path for a flow. Further, we will devise techniques for detecting this behavior in operational networks. To address this problem, we will need to augment our detection tools with additional measurement systems that fingerprint and isolate the network location of each device along a path. For example, if we can send traffic such that at most one middlebox is on the path to the destination, we can identify each middlebox location and understand its configuration. Then we can use this information and our models to understand how a these devices collectively result in unintended performance.

References

[1] M. Dischinger, M. Marcon, S. Guha, K. P. Gummadi, R. Mahajan, and S. Saroiu. Glasnost: Enabling end users to detect traffic differentiation. In *Proc. of USENIX NSDI*, 2010.

[2] FCC. Order 6712-01: Protecting and promoting the open internet. `https://s3.amazonaws.com/public-inspection.federalregister.gov/2015-07841.pdf`, April 2015.

[3] J. Hui, K. Lau, A. Jain, A. Terzis, and J. Smith. How YouTube performance is improved in T-Mobile network. `http://velocityconf.com/velocity2014/public/schedule/detail/35350`.

[4] A. M. Kakhki, A. Razaghpanah, R. Golani, D. Choffnes, P. Gill, and A. Mislove. Identifying traffic differentiation on cellular data networks. In *ACM SIGCOMM Poster & Demo Session*, 2014.

[5] A. Rao, J. Sherry, A. Legout, W. Dabbout, A. Krishnamurthy, and D. Choffnes. Meddle: Middleboxes for increased transparency and control of mobile traffic. In *Proc. of CoNEXT 2012 Student Workshop*, 2012.