An Empirical Evaluation of Deployed DPI Middleboxes and Their Implications for Policymakers

David Choffnes* Phillipa Gill† Alan Mislove*
*Northeastern University †University of Massachusetts Amherst

Abstract

Middleboxes are commonly deployed to implement policies (e.g., shaping, transcoding, etc.) governing traffic traversing ISPs. While middleboxes may be used for network management to limit the impact of bandwidth-intensive applications, they may also be applied opaquely to limit access to (or degrade) services that compete with those offered by the network provider. Without regulation or accountability, such practices could be used to raise the barrier to entry for new technologies, or block them entirely. Further, by breaking end-to-end system design principles, these practices can have negative side-effects on reachability, reliability and performance.

This paper presents evidence of deployed middlebox-enabled policies that provide differential service to network applications affecting subscribers of T-Mobile US, Boost Mobile, and others. We used rigorous controlled experiments and statistical analysis of the performance of popular online services to identify traffic differentiation. The observed policies include throttling bandwidth available to video and audio streaming, transcoding video, and selectively zero-rating traffic such as video and music streaming. Such policies may violate the "No Throttling" and/or "No Unreasonable Interference" provisions of the Open Internet Order [15] (OIO), and potentially violate rules in different jurisdictions. Some of these policies were not transparent to consumers and/or were presented in misleading ways, violating the transparency requirement of the OIO. We recommend that providers concerned about traffic loads use application-agnostic techniques to throttle, thus meeting the "reasonable network management" clause of the OIO. Such policies are also easy for consumers to understand, thus providing better transparency.

We find that the observed policies are implemented using deep packet inspection (DPI) and simple text-matching on the contents of network traffic, potentially leading to misclassification. We validate that misclassification occurs, causing unintentional zero-rating or throttling. For example, video-specific policies can arbitrarily apply to non-video traffic, providing another example of "Unreasonable Interference" barred in the OIO. In fact, we show that current approaches to implementing network management policies are fundamentally vulnerable to unintentional behavior; i.e., the DPI-based approach to network management cannot guarantee 100% accuracy. We recommend that the specific implementations of DPI-based throttling be made public to improve transparency. Further, we recommend that policymakers and network operators adopt alternative rules and approaches to network management that avoid such flaws and vulnerabilities.

Last, network management policies currently lack auditing provisions, and we argue that this hinders enforcement and compliance with rules. Further, network providers' policies evolve over time, requiring constant vigilance. We recommend that regulators incorporate auditing technologies such as those presented in this work as part of future policies.

1 Introduction

The rise in popularity of bandwidth-hungry applications such as video streaming and the popularity of resource-constrained mobile networks has reignited discussions about how network traffic associated with

different applications is treated (or mistreated) by ISPs. To manage scarce network resources, ISPs use *middleboxes* such as transparent proxies, shapers, and transcoders to selectively enforce *network policies* (e.g., compressing and caching static Web content, limiting transfer rates for video streaming, and reducing image quality). While middleboxes may be used for reasonable network management (e.g., to limit the impact of bandwidth-intensive applications), they may also be applied opaquely to limit access to (or degrade) services that compete with those offered by the network provider.

Without regulation or accountability, such practices could be used to raise the barrier to entry for new technologies, or block them entirely. Further, by breaking end-to-end system design principles, these practices can have negative side-effects on reachability, reliability and performance. For example, in recent peering disputes Comcast refused to give extra capacity to the Netflix video streaming service without additional payment from Netflix [16]. Further, our recent work identifying differentiation in cellular networks [19] demonstrated that Boost Mobile transcoded YouTube videos to lower bitrates (up to a 66% reduction).

Despite the importance of this issue, the impacts of policies on specific applications—and the Internet as a whole—are poorly understood. For example, previous efforts that attempt to detect traffic differentiation [13,31,37,38] are limited to specific types of application traffic (e.g., peer-to-peer traffic [13]), or operate in a manner that is oblivious to the specific mechanisms used for traffic differentiation [31,38]. In parallel, several studies attempt to characterize proxies in mobile networks [12,32,36], but do so using an ad-hoc set of measurement tests that do not necessarily reflect traffic generated by real applications. These limitations stem from the wide variety of applications (some of which are closed source) that make extending measurements to arbitrary applications challenging, and the closed nature of middlebox implementations. Indeed, their closed nature means that external observers struggle to detect if middleboxes exist in networks, what exactly these devices do, and what is their impact on traffic. What little we do know is concerning. Glasnost [13] and others [28,31,37] identified how ISPs use devices to throttle Internet traffic, Cogent admits to differentiating Netflix traffic [26], and recent work by Google and T-Mobile [17] indicate that interactions between middleboxes can severely degrade performance.

This paper presents evidence of deployed middlebox-enabled policies that provide differential service to network applications affecting subscribers of T-Mobile US, Boost Mobile, and others. We used rigorous controlled experiments and statistical analysis of the performance of popular online services to identify traffic differentiation. The observed policies include throttling bandwidth available to video and audio streaming, transcoding video, and selectively zero-rating[1] traffic such as video and music streaming. Such policies may violate the "No Throttling" and/or "No Unreasonable Interference" provisions of the Open Internet Order (OIO), and potentially violate rules in different jurisdictions. Some of these policies were not transparent to consumers and/or were presented in misleading ways, potentially violating the transparency requirement of the OIO. We recommend that providers concerned about traffic loads use application-agnostic techniques to throttle, thus meeting the "reasonable network management" clause of the OIO. Such policies are also easy for consumers to understand, thus providing better transparency.

We find that the observed policies are implemented using *deep packet inspection*[2] (DPI) and simple text matching on the contents of network traffic, potentially leading to misclassification. We validate that misclassification occurs, causing unintentional zero-rating or throttling. For example, video-specific policies can arbitrarily apply to non-video traffic, providing another example of "Unreasonable Interference" barred in the OIO. In fact, we show that current approaches to implementing network management policies are fundamentally vulnerable to unintentional behavior; i.e., the DPI-based approach to network management cannot guarantee 100% accuracy. We recommend that the specific implementations of DPI-based throttling be disclosed to improve transparency. We recommend that policymakers and network operators adopt alternative rules and approaches to network management that avoid such flaws and vulnerabilities.

Last, network management policies currently lack auditing provisions, and we argue that this hinders enforcement and compliance with rules. Further, network providers' policies evolve over time, requiring constant vigilance. We recommend that regulators incorporate auditing technologies such as those presented in this work as part of future policies.

[1] Zero rating refers to the practice of not charging Internet traffic against a subscriber's monthly quota.

[2] The Internet runs on devices such as routers and switches that inspect network packets to determine how they should be forwarded; however, DPI devices additionally inspect other data in network packets. This includes payloads such as web pages, video streams, and messages that establish secure, encrypted connections.

2 Background

The Internet provides a uniform way to connect content providers and consumers across disparate, independently-operated networks. A key factor enabling its success is that the network is typically neutral with respect to the packets it carries [10]; i.e., networks do not discriminate against or otherwise alter certain types of traffic except for lawful purposes (e.g., blocking illegal content). In this section, we discuss the history of middleboxes that break this assumption, previous attempts to detect them, and how there is a need for a more systematic approach.

Detecting policies and their implications. The regulatory framework surrounding network management policies is rapidly changing; in June 2015, the FCC transitioned from imposing few restrictions on how mobile providers manage their network to new rules that prohibit many forms of differentiation [15]. More recently, the FCC is considering revisiting and/or removing such rules. Outside the US, the Body of European Regulators for Electronic Communications (BEREC) adopted guidelines for National Regulatory Authorities on the implementation of net neutrality regulations in 2016 [8], and member nations such as ARCEP are actively investigating ways to enact corresponding rules. In 2017, the Canadian Radio-television and Telecommunications Commission (CRTC) incorporated network neutrality principles into a recent decision forbidding zero-rating (and throttling) of specific services [11]. In this environment of evolving regulations, it is important to monitor ISP policies in practice, both to inform regulators and enforce policies.

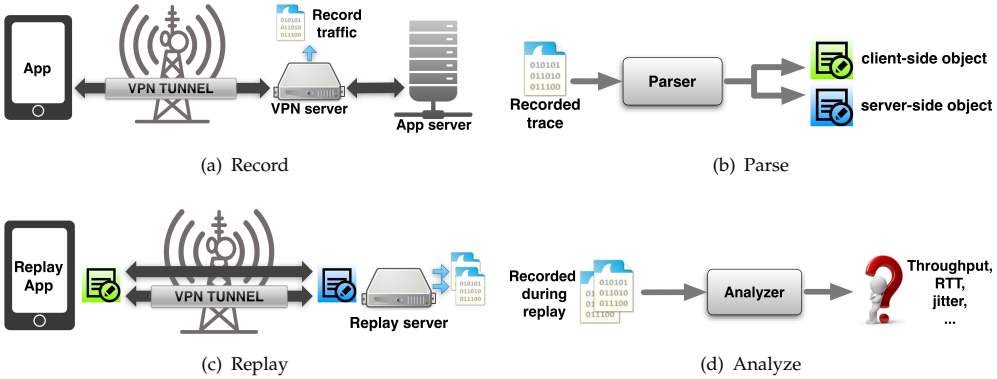
Incidents such as Comcast's blocking of BitTorrent [29] led to early research efforts to study network policies. Several studies focus on a small set of applications or flows known to be subject to classification, and design tests to detect policies for them. For example, Glasnost focuses on BitTorrent [13], and NetPolice [37] uses five applications. Other related studies detect transparent proxies, and conduct tests to identify the presence of known proxy behaviors [3, 21, 30, 33, 35].

Another approach ignores the classification problem and focuses on detecting the existence of policies for an arbitrary class of traffic. NANO [31] proposes using Bayesian analysis to detect policies, while Zhang et al. [38] focus on identifying when it is feasible to detect differentiation and how to isolate it. Other related work focuses on identifying performance in ISPs [2, 14, 23, 25] but does not detect an ISP's policies. Without ground-truth information about how polices are implemented in operational networks, and how they impact traffic in a production environment, it is challenging to tell how well these techniques work.

A number of studies focus on the problem of detecting transparent proxies [21, 27, 34]. These studies require a client to run a Java-based applet (generally not supported on smartphones), and control of the destination server for measurement traffic. Other related work extends this approach to the mobile environment [12, 32], but these tools work only for Android devices and require rooted phones for some of the tests. All of the above projects make strong assumptions about what kind of traffic might enable middlebox detection, and none provide a systematic way for doing so for arbitrary policies, middleboxes, and/or applications.

Limitations of assumptions in prior work. We have built a testbed with operationally deployed middleboxes to understand whether assumptions in previous work hold in practice. Using a small set of initial experiments on a packet shaping device, we find that many key assumptions are violated:

- **DPI middleboxes include traffic classifiers that identify hundreds of apps/services.** One of our devices lists more than 700 applications that it uniquely identifies. Thus, focusing on one (or a small number) of apps is insufficient.
- **Sending packets on random ports is not sufficient to detect policies.** Glasnost's [13] detection approach assumes that traffic sent on random ports will not be affected by middleboxes, but traffic traffic sent on standard application ports (e.g., port 80 for HTTP) will. One of our devices generally classifies traffic on random ports as Peer-to-Peer traffic (P2P), which itself is often subject to differentiation. Thus, previous techniques for identifying policies are likely to fail.
- **Previously used statistical tests to detect differentiation are unreliable.** In a testbed environment with a commercial shaper, we replay the same flow multiple times, with and without shaping enabled. Using the KS Test as done in NetPolice [37] to determine whether two samples of throughput are



(a) Record (b) Parse

(c) Replay (d) Analyze

Figure 1: System overview: (a) Client connects to VPN, that records traffic between the client and app server(s). (b) Parser produces transcripts for replaying. (c) Client and replay server use transcripts to replay the trace, once in plaintext and once through a VPN tunnel. Replay server records packet traces for each replay. (d) Analyzer uses the traces to detect differentiation.

different (as evidence of differentiation), we find that our results have significant false positive and false negative rates [19]. The key problem is that such statistical tests focus on distributions of values, but not *differences* in values between two distributions. There is thus a need to develop new tests that are tuned with ground truth validation.

Key challenges for measurements of DPI devices. This methods described in this paper tackle three key challenges that have held back prior work on measuring traffic differentiation: (1) the inability to test arbitrary classes of applications, (2) a lack of understanding of how DPI devices work in practice, and (3) the limited ability to measure mobile networks (e.g., cellular and WiFi) from end-user devices such as smartphones and tablets.[3] Designing methods that can measure traffic differentiation in mobile networks (in addition to well-studied wired networks) presents unique challenges, as the approaches must work with highly variable underlying network performance and within the constraints of mobile operating systems. By addressing these challenges, we provide tools that empower nontechnical users (e.g., subscribers and regulators) to identify differentiation mobile and fixed-line networks, and use data gathered from these tools to understand differentiation in practice.

3 Approach

Detecting differentiation using "record and replay." When traffic differentiation occurs, it is often the case that one application (or class of applications, such as video) receives different performance than others. Typically an ISP inspects the contents of network traffic to determine the application (e.g., looking for "netflix.com" to identify Netflix). Thus, to test whether differentiation occurs in an ISP, it is essential to use network traffic that comes from the targeted applications.

We developed a technique called "record and replay" that does just this (Figure 1).[4] First, we record network traffic when using real applications such as YouTube and Netflix. Next, we use our own software that generates the same traffic between clients in an ISP and servers that we provide. If there is differentiation in the network, this "exposed" replay traffic will also be affected in the same way as application-generated traffic. Last, we send the same exact amount of network traffic as a "hidden" replay, where we encrypt the traffic so its contents are hidden from the ISP. If there was different performance for the exposed replay than the hidden one, then we can conclude there is differentiation in that ISP for that application.

A key challenge is to reliably determine that differences in performance between "exposed" and "hidden" are due to differentiation, as opposed to other confounding factors such as normal bandwidth varia-

[3] Note that the methods and many of the findings described in this paper were previously published by the authors [18, 19, 22]. This working paper is a summary of those results. More technical details behind the work can be found in the citations.

[4] For more details, please see our technical paper on the topic [19].

tions over wireless cellular technologies. A second challenge is that we must limit ourselves to identifying differentiation that is likely to have a significant impact on the user experience, as opposed to bandwidth caps that are higher than the maximum bitrate for a video. Prior work did not address these issues: Glasnost [13] tests for bandwidth caps but does not compare them with the bandwidth consumed by an application in practice, while NetPolice [37] uses an approach (KS Test) that is robust to this but might flag cases where differences in performance are so small that they do not affect applications. To address this latter problem, we proposed an *Area Test* that accounts for the degree of differentiation detected: our test concludes there is differentiation if the KS Test detects differentiation *and* the normalized area between the performance curves is greater than a threshold.

Identifying DPI matching rules. The previous steps identify *when* differentiation occurs, but not *why* certain traffic is selected for differentiation. Importantly, IP traffic does not contain any identifiers that reliably indicate what kind of application or service is being used. Instead, ISPs that want to selectively throttle, zero-rate, or otherwise differentiate against an application using their network often must resort to using DPI devices that inspect the contents of network traffic for clues as to what application it corresponds to.

We found empirically that these clues often take the form of text (e.g., in the Netflix example above) or specific byte patterns (e.g., to indicate VoIP traffic). DPI devices search the contents of network traffic and attempt to match them to one or more text (or byte) patterns. We refer to these as *matching rules*. For example, "*Packet contains 'netflix'*" might be a simple matching rule for Netflix traffic.

Note that the example in the previous paragraph can be problematic. For example, if a user is downloading a Web page or an e-book with the term "netflix" in it, it would be (incorrectly) classified as Netflix traffic. Thus, it is essential to understand how DPI devices select matching rules for certain applications and what are their implications for misclassification. Because we do not know the matching rules in advance, we developed an approach that allows us to reveal them empirically.[5]

The first step in understanding matching rules is determining which portions of network flows contain content that matches. In the base case (nothing is known about matching rules), we conduct a binary search where we replace half of the flow with random bytes and observe its effect on classification. Our assumption is that random bytes are very unlikely to match any classification rules.[6] If the traffic is no longer classified as the recorded application, we then identify a more specific region in which the method will be applied on—namely, the half of bytes that triggered the change. To do this, we first revert the bytes in that region back to their original content, then repeat the process of changing one half of the bytes at a time in that region. If both halves triggered a change, then we identify both halves as triggering the matching rules. Once we identify portions of network flows that trigger matching rules, we conduct more extensive tests by modifying each byte of a packet, one at a time, until we find the set of bytes that affect classification. These bytes comprise the field(s) used for matching.

The second step is generalizing the matching fields into rules. This is the difference between a matching field being "`Host: someprefix.netflix.com`" and "First look for `Host:`, then check whether the line ends with `netflix.com`." To reveal the precise matching rule, we conduct tests that randomize and otherwise alter subsets of content in each field until the minimal set of matching content is identified.

Limitations. Our approaches make several assumptions that we discuss briefly here. First, we assume that we can generate "hidden" traffic that is not subject to differentiation. We have yet to find a network where we could not do this. Second, we assume that an ISP determines when to apply differentiation based on DPI, not destination IP address. This is generally reasonable because IP addresses are not strongly tied to differentiated services; for example, most video providers serve content using CDNs that use servers covering a wide range of IP addresses and hosting a wide range of other non-differentiated traffic. Third, we assume that we can detect differentiation by measuring and comparing network traffic and performance. In practice, we find this assumption to be true, but it remains an open question how well it will hold in the future. Last, we assume that DPI middleboxes identify targets of differentiation using text or byte patterns in IP packets, but not packet sizes or timing. Again, this assumption has proven to be valid in all of our case studies, but may change in the future or in other networks.

[5]For more details, please see our technical paper on the topic [22].

[6]We validated this assumption by running 1,000 flows, each with different random bytes (from 100 to 1000 bytes) on port 80, all of which were classified as the same generic "HTTP" class.

4 Case Studies and Implications

We now describe DPI-based policies that we identified using the approach from the previous section, and discuss their implications for subscribers and content owners. These cases come from experiments conducted between 2015 and 2017, and the year of the most recent experiment identifying the behavior is listed for each case.

4.1 DPI-based throttling and/or zero-rating

T-Mobile US (2016, *Binge On* and Music Freedom) In the fall of 2015, T-Mobile US announced their *Binge On* program, offering zero-rated video for subscribers with a cap on download speeds. At the time, T-Mobile claimed that [4]:

> *Binge On optimizes video quality for smartphone screens. It provides a great DVD-quality experience (typically 480p or better) for all detectable video, which can minimize buffering and maximize quality while using a fraction of the data...*

They also claimed to zero-rate such traffic if consumers and content providers opted in. We investigated their implementation in detail in early 2016 [18] and summarize key findings below.[7]

DPI implementation. We make the following observations about the *Binge On* implementation.

- *The* Binge On *rate limit is ≈1.5 Mbps.* This rate is sufficient for 480p videos on YouTube and "low quality" Netflix streaming, but is often below requirements for higher video quality (e.g., 720p for YouTube requires 1.5–4 Mbps [5] and Netflix SD quality video requires 3 Mbps [6]). Google's VP9 codec supports 720p at bitrates lower than 1.5 Mbps, but the only smartphones that supported it at the time were select Android ones. Thus, T-Mobile's claim of supporting "480p or better" [4] video quality is technically true, but in practice support for "better" depends on the device and video service. Interestingly, when *Binge On* is enabled, YouTube selects medium (360p) quality, lower than the 480p specified by T-Mobile.
- *The* Binge On *infrastructure does not "optimize" video.* T-Mobile's CEO claimed that "*Binge On* includes a proprietary technology to not only detect the video stream, but select the appropriate bit rate to optimize to the mobile device" [1]. Our differentiation detection methodology trivially reveals whether an ISP modifies content, e.g., for optimization. We found no modification to our replay content, nor any evidence that *Binge On* behavior changed in response to the device we used (smartphone, or laptop).
- Binge On *is implemented using policing.* There is low jitter and high retransmission rates when *Binge On* is enabled. This indicates a token bucket with a small (or no) queue which results in packets being dropped when there are no tokens available.[8]
- Binge On*'s rate limit is cumulative for all flows from a single SIM.* If a customer streams simultaneous *Binge On*-eligible videos (via tethering), the average throughput per stream will be lower than 1.5Mbps.
- Binge On *traffic was typically zero-rated while traffic from services not participating in* Binge On *was charged.*When we tested with replays of video traffic from providers not participating in *Binge On*, the data used was charged against our data plan. Thus, at the time of our experiments, YouTube traffic was throttled to 1.5 Mbps *and we were charged for the data*, whereas Netflix was similarly throttled and there was no data charged.[9] We conducted the same experiments for other *Binge On* participants (e.g., HBOGo, ShowTime, and Hulu) and non-participants (e.g., Vimeo and Veoh), and found identical results. We also recorded and replayed other types of traffic (e.g., image download), and observed no rate limits imposed on them.

[7] Most of the content for this case study comes from our publication on the topic [18].

[8] Our replays do not adapt bitrates; rather, video traffic is replayed at the same bitrate it was recorded, which may be higher than supported by BingeOn.

[9] Note that list of services participating in *Binge On* changes over time (e.g., YouTube joined *Binge On* just after our study), but we confirmed this behavior is true for every tested video service not participating in *Binge On*.

Application	Detection criteria	
	Binge On/Music Freedom*	Video
Netflix	Specific GET arguments and the term "Netflix"	Same as *Binge On*
HBOGo	`Host` header ends with "hbogo.com"	`Content-Type` header (video/mp2t)
ShowTime	`Host` header ends with "showvodhls.edgesuite.net**"	`Content-Type` header (video/mp2t)
Hulu	`Host` header ends with "hulu.com"	`Content-Type` header (video/mp2t)
Amazon Video	`Host` header ends with "amazonvod.loris.llnwd.net**"	`Content-Type` header (video/mp2t)
Veoh	Not part of *Binge On*	`Content-Type` header (video/mp4)
Vimeo	Not part of *Binge On*	Unknown***
YouTube (HTTP)	`Host` header ends with "googlevideo.com"	`Content-Type` header application/octet-stream
YouTube (HTTPS)	Server name in the SNI ends with "googlevideo.com"	Same as *Binge On*
Spotify*	"Spotify" in `Host` and `User-Agent` headers	N/A
Pandora*	`Host` header ends with "p-cdn.com"	N/A

Table 1: Summary of how each app is detected by T-Mobile. If a flow matches a *Binge On* app, the traffic will be zero-rated and throttled. Otherwise, it will check for video signatures and throttle such traffic (but not zero-rate) if it matches. **Music Freedom*, is a program similar to *Binge On*, which zero-rates music streaming apps [7]. **edgesuite.net (run by Akamai) and LLNWD.net (run by Limelight) are CDNs serving ShowTime and Amazon videos. It is possible that these providers use other servers/CDNs too, hence different host names exist that result in zero-rating. ***We did not observe rate-limiting, which we use to reverse engineer video detection criteria. We suspect that *Binge On* classifiers are not properly configured for Vimeo.

- *We found that* Binge On *behavior is not entirely consistent over time.* We encountered a small number of cases where a *Binge On*-participating service's traffic was not zero-rated. These cases were transient, suggesting they are due to reasons such as buggy or overloaded infrastructure that supports *Binge On*.

We further reverse engineered the matching rules that T-Mobile's DPI device used for identifying *Binge On* and non-*Binge On* video traffic. Table 1 summarizes our findings for several popular video streaming services. A key take-away is that *Binge On* uses a DPI device that matches text in network traffic to detect video and *Binge On*-eligible traffic. As we discuss below, this means that their policy can easily be erroneously applied to traffic. Our key findings are as follows:

- ***Binge On* uses matching rules on Host, Content-Type, and HTTPS Server Name Indicator (SNI) fields.** The left two columns in Table 1 show that *Binge On* uses simple text-based matching rules. In the case of encrypted (HTTPS) traffic, the contents of network traffic (e.g., the Content-Type) are hidden from the DPI device. In these cases, the DPI devices relies on SNI, which indicates the domain name (e.g., `googlevideo.com` being visited. This is a poor proxy for content type. Classification first prioritizes the signature for *Binge On* apps; if there is no app-specific match then *Binge On* uses `Content-Type` signatures to detect non-*Binge On* video streams.
- ***Binge On*'s matching rules are brittle.** Once T-Mobile successfully matches *Binge On*-specific text in its classifier, it ignores all other fields that might support *or contradict* the classification. For example, at the time of our measurements HTTP traffic to `non-video-content.googlevideo.com` would be classified as YouTube video and throttled, even if (as the hypothetical name suggests) the traffic does not contain video.
- **IP addresses and port numbers have no impact on classification.** Our replay experiments use different IPs from the recorded services, and we also experimented with changing port numbers. We find that classification does not depend on the port, e.g., a request with `Host: hbogo.com` and a video `Content-Type` to port 55555 (as opposed to the standard 80) on our replay server is still detected by T-Mobile as HBOGo.

Application	Music Unlimited matching rule	Throttled to 128 Kbps?
Google Play Music	SNI field contains "sj.sj.googleusercontent.com"	No
Spotify	`Host` header contains "spotify.com.edgesuite.net"	No
Stingray	`Host` header contains "private-mediafile.galaxie.ca"	No
Bandcamp	`Host` header contains "p4.bcbits.com"	No

Table 2: Summary of how each music streaming app is detected by Videotron.

- ***Binge On* uses app-specific strings to throttle without zero-rating.** The "YouTube (HTTPS)" row in Table 1 shows that *Binge On*-ineligible video streaming apps[10] are specifically targeted for throttling, in addition of general rules for all videos.

Implications. T-Mobile's *Binge On* implementation has a number of important implications for policy-makers and regulators.

- *There is a gap between what was disclosed and what was implemented.* Claims about supporting "480p or better" and "optimizing" video are unsubstantiated and inconsistent with the reality of the simple 1.5 Mbps rate limit on video-transfer rates, which is shared by all users connecting via the same SIM.
- *The zero-rating implementation led to unexpected and potentially unfair practices.* For example, YouTube was throttled but not zero-rated, while Netflix was throttled and zero-rated—this was advantageous for Netflix when compared with YouTube. There were also inconsistencies in how *Binge On* traffic was billed over time.
- *The DPI implementation is brittle.* T-Mobile's program relied on matching text in network traffic to identify video services. These could lead to false positives (when the matching text is not video-specific) and false negatives (when a video stream does not contain any matching text). The problem will only get worse as encryption becomes more common and less information about an application is provided in plaintext. We also found that this brittle implementation led to inconsistencies with disclosures, e.g., Vimeo traffic is not throttled or zero rated even though it is detectable video.

Videotron (2016) Videotron, a mobile network provider in Canada, provided zero-rated music streaming (dubbed *Unlimited Music*) as part of their offering in 2016. In collaboration with Fenwick McKelvey, we measured their implementation. Our results were summarized in a CRTC hearing on October 31, 2016.[11] The disclosure for Unlimited Music "excludes the downloading of songs and other non-musical content" and "allows a maximum flow of 128 Kbps."

DPI implementation. We conducted experiments that were nearly identical in nature to those conducted for the *Binge On* case study, but this time focused on nine streaming music/audio services. We found that Videotron decided to zero-rate traffic based on DPI matching rules for HTTP headers and HTTPS handshakes for the following apps: BandCamp, Stringray, Spotify, Google Play Music and DigitallyImported. Interestingly, we found no evidence of throttling. Statistically speaking, zero-rated and charged packets get equal access to available bandwidth.

Implications. Videotron's Music Unlimited service is no longer allowed under a recent CRTC ruling. That said, we can still identify interesting implications for this case.

- *Videotron's matching rules were not sufficient to detect all music services.* As a result, there was a potential for some services to unfairly benefit from zero rating.
- *Contract terms and disclosures were inconsistent with empirical observations.* Videotron surprisingly never implemented throttling as specified in their contract. When responding to Dr. McKelvey's testimony to the CRTC, Videotron claimed that their DPI "solution examines different characteristics of flow, such as sequence and packet size, to determine the most likely application." In our tests, we replayed

[10] Before YouTube was added to *Binge On*.

[11] `http://www.crtc.gc.ca/eng/transcripts/2016/tt1031.htm`

ISP	YouTube	Netflix	Spotify	Skype	Viber	Hangouts
Verizon	M	M	M	-	-	-
T-Mobile	-	-	-	-	-	-
AT&T	F	F	F	-	-	-
Sprint	M/P	M/P	M/P	-	-	-
Boost	M	M	M	-	-	-
BlackWireless	60%	-	-	-	-	-
H2O	37%*	45%*	65%*	-	-	-
SimpleMobile	36%	-	-	-	-	-
NET10	P	P	P	-	-	-

Table 3: Shaping detection results per ISP in our dataset, for six popular apps: YouTube, Netflix, Spotify, Skype, Viber, and Google Hangouts. When shaping occurs, the table shows the difference between average throughput (%) we detected. A dash (-) indicates no differentiation, (F) means IP addresses changed for each connection, (P) means a "translucent" proxy changed connection behavior from the original app behavior, and (M) indicates that a middlebox modified content in flight between client and server. *For the H2O network, replays with random payload have better performance than VPN and exposed replays, indicating a policy that favors non-video HTTP over VPN and streaming video.

traces of music streaming where we maintained packet sizes and sequence, but modified packet contents corresponding to matching rules. These tests were not zero-rated, indicating that such a solution is unlikely to be deployed.

Verizon Wireless (2017) A recent article [9] identified that Verizon throttles video streaming to 10 Mbps on all postpaid plans. This disclosed behavior is sufficient for HD video streams, and thus it unlikely affects video streaming quality for mobile devices. We ran independent tests to better understand this new policy. We used a prepaid plan, which included the following disclosure regarding caps on download speeds: "Functionality of some data applications, like streaming video or audio, may be impacted. Video streams up to 480p."

DPI implementation. We found that Verizon's DPI device detects and throttles video in a nearly identical way to T-Mobile's *Binge On* program. Similar to *Binge On*, Verizon's DPI device detects many video streaming services, but not Vimeo. Unlike *Binge On*, Verizon throttles video streaming for prepaid plans to 2 Mbps, which is substantially lower than the 10 Mbps reported for (presumably) postpaid plans. Based on our *Binge On* study, we believe that the claim of "video streams up to 480p" is accurate.

Implications. The implications for this streaming service are nearly identical to those for *Binge On*. Like *Binge On*, not all services are affected by the cap, including services of the same type (e.g., video streaming from Vimeo). Unlike *Binge On*, we find the disclosure of video quality to be accurate.

Other cases (2015) We identified several cases of DPI-based differentiation on video traffic in our study of cellular networks in January–May, 2015 (see Table 3). At the time, we did not have the ability to reverse engineer matching rules, so we describe only the performance differences observed.

Implications. The fact that BlackWireless and SimpleMobile were selectively throttling YouTube traffic indicates that certain content providers were unfairly discriminated against when it comes to offering high-quality video to their subscribers. Note that all of these behaviors ceased after the FCC's Open Internet Order was passed, according to tests in August, 2015. This suggests that the OIO successfully addressed unfair ISP practices that existed prior to the rules.

4.2 Content injection

Verizon Wireless (2015) Verizon inserted so-called tracking "supercookies" into their subscribers' network traffic [24], to the surprise of many subscribers. Specifically, Verizon injected into Web traffic an HTTP header with a subscriber-unique identifier that could be used by third parties to tie network activity back to an individual. We also observed this behavior in our experiments during our test on March 5, 2015.

Verizon was fined by the FCC for this behavior days later [20] and no longer injects this content into web traffic without user permission.

4.3 Transcoding

Boost Mobile (2015) We found that Boost Mobile (an MVNO owned by Sprint) transcoded video traffic into a lower quality bitrate, and also cached the transcoded content in their network. This was problematic in two ways. First, the content was not owned by Boost and it is not clear that they had the rights to change it in flight without YouTube's permission. Additionally, Boost ignored directives in Google's HTTP headers that explicitly indicated that content exchanged with clients should not be cached.

Sprint (2015) We found that Sprint transcoded certain images to a lower quality level than originally encoded, as originally discovered by Xu et al. [36]. This has potential benefits to subscribers in the sense that the downsampled images were smaller in terms of the file size, and thus could save bandwidth and page-loading time. However, to the best of our knowledge the policy was not disclosed to users, they were given no way to opt out, and it was not evenly applied to all images. For example, we found that this practice only applied to JPEG images whose original file size is 500 KB or less.

4.4 Different performance for "random" traffic

We found that H2O gave higher performance to network traffic that was not streaming audio/video or VPN traffic. This is potentially problematic because new applications could use knowledge of this behavior to obtain better performance than their competitors by modifying their packet contents to look like "random" traffic.

5 Recommendations

Based on our findings, we make the following recommendations for improving policies governing ISP practices.

Integrating empiricism into regulation. Regulations should be designed such that compliance can be measured empirically. By building specific network tests for compliance into regulations, both network providers, consumers, and regulators can immediately and automatically determine the legality of a network provider's policies. This could also vastly simplify and streamline processes for filing and responding to complaints.

Auditing. Alongside regulations that incorporate empirical measures of compliance, there is need for auditing tools that allow regulators to test network providers even when no complaints have been filed. Ideally, such auditing tools should also be made available to the public and should be easy-to-use for non-technical consumers, thereby improving transparency and reducing opportunities for network providers to "game" any one auditing client.

DPI disclosure. Our research showed that DPI devices remain prevalent in broadband access networks, but there is a gap between the policies that providers disclose (e.g., throttle all video traffic) and the DPI implementations (e.g., throttle only video traffic that matches a set of incomplete rules). We recommend that any deployed DPI devices (or similar traffic classification devices) must be disclosed both to consumers and regulators. The classification criteria (i.e., the matching rules) must also be public so that affected parties can evaluate their impact. Without such details, current disclosures are at best imprecise and at worst misleading.

Analysis of collateral damage. We discussed and demonstrated that DPI devices can misclassify traffic, leading to unintended and undisclosed differentiation. In some cases this may be benign, e.g., if a small fraction of content is unintentionally zero-rated. In other cases, this can lead to a poor subscriber experience and/or unfair practices, e.g., when videoconferencing traffic is unintentionally throttled. We recommend that regulators and researchers develop a systematic way to understand potential benefits and harms of incorrect classification.

Reasonably managing network loads. All of the throttling policies we observed took effect no matter the location or time of day that we tested. However, it is well known that the bandwidth demand on ISPs follows diurnal patterns, and that bottlenecks in such networks tend to be isolated geographically (e.g., to a cell tower) and transient (i.e., lasting seconds or minutes). Thus, there is a disconnect between the need for managing traffic demands that exceed capacity (which are isolated) and the policies used in practice (which are always on, everywhere). There is a simple alternative to the current approach: give every active subscriber an equal share of the available bandwidth when demand exceeds capacity. This is application-agnostic, meaning only the subscriber's services that contribute to excessive demand are affected, instead of singling out a specific application or application class. It also would occur only when demand exceeds capacity, meaning that subscribers of a well provisioned ISP could enjoy all Internet services without artificial limits, the vast majority of the time.

Zero rating. The practice of zero-rating certain classes of network traffic is becoming increasingly common in today's mobile network providers. While this can potentially benefit all parties (subscribers, network providers, and content providers), it can also potentially become a vehicle for unfair business practices (e.g., zero-rating content owned by the network provider). We recommend an interdisciplinary, empirically informed effort to understand of zero rating and its impact on consumer choice and content providers, particularly when coupled with throttling.

6 Conclusion

Recent years have seen rapid evolution in the products that network providers advertise to consumers, and the corresponding network policies that apply to subscriber traffic. Likewise, regulators have imposed new rules that govern which of these policies are legal. However, there are gaps between what network providers advertise and what is deployed in practice, typically using DPI middleboxes. Further, regulators often lack the tools or regulatory detail to audit network providers when such products may cause harm to subscribers or content providers.

In this work, we discussed ongoing research that addresses these gaps—both by providing tools that identify exactly what policies are in place, and by discussing the implications of these policies. We identified a wide variety of practices that target specific applications or types of applications, and discussed how such practices may be discriminatory to the benefit or detriment of subscribers and content providers. We also found that such practices change over time, motivating the need for continuous auditing of network providers. We made several recommendations for policymakers based on our observations; namely, incorporating rigorous measurements into telecom regulation and use them to audit providers, requiring network providers to disclose their precise DPI-based policies, and studying the impact of collateral damage and zero-rating practices. Finally, we proposed a simple, effective approach to reasonable network management that raises none of the neutrality concerns associated with throttling and/or zero-rating a subset of applications.

References

[1] https://twitter.com/JohnLegere/status/685201130427531264.

[2] Mobilyzer. http://www.mobilyzer-project.mobi.

[3] Neubot – the network neutrality bot. http://www.neubot.org.

[4] T-Mobile BingeOn. http://www.t-mobile.com/offer/binge-on-streaming-video.html.

[5] Live encoder settings, bitrates and resolutions. https://support.google.com/youtube/answer/2853702, Mar. 2016.

[6] Netflix internet connection speed recommendations. https://help.netflix.com/en/node/306, Mar. 2016.

[7] T-Mobile Music Freedom. http://www.t-mobile.com/offer/free-music-streaming.html, Mar. 2016.

[8] BEREC. BEREC guidelines on the implementation by national regulators of european net neutrality rules. http://berec.europa.eu/eng/document_register/subject_matter/berec/regulatory_best_practices/guidelines/6160-berec-guidelines-on-the-implementation-by-national-regulators-of-european-net-neutrality-rules, August 2016.

[9] J. Brodkin. Verizon accused of throttling netflix and youtube, admits to "video optimization". https://arstechnica.com/information-technology/2017/07/verizon-wireless-apparently-throttles-streaming-video-to-10mbps/, July 2017.

[10] D. Clark. Network neutrality: Words of power and 800-pound gorillas. *International Journal of Communication*, 2007.

[11] CRTC. Telecom regulatory policy CRTC 2017-104. http://crtc.gc.ca/eng/archive/2017/2017-104.htm, April 2017.

[12] G. Detal, B. Hesmans, O. Bonaventure, Y. Vanaubel, and B. Donnet. Revealing middlebox interference with tracebox. In *Proc. of IMC*, 2013.

[13] 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.

[14] FCC announces "Measuring Mobile America" program. http://www.fcc.gov/document/fcc-announces-measuring-mobile-america-program.

[15] FCC. Protecting and promoting the open internet. https://www.federalregister.gov/articles/2015/04/13/2015-07841/protecting-and-promoting-the-open-internet, April 2015.

[16] S. Higginbotham. The Netflix–Comcast agreement isn't a network neutrality violation, but it is a problem. http://gigaom.com/2014/02/23/the-netflix-comcast-agreement-isnt-a-network-neutrality-violation-but-it-is-a-problem/, February 2014.

[17] 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.

[18] A. M. Kakhki, F. Li, D. R. Choffnes, E. Katz-Bassett, and A. Mislove. BingeOn under the microscope: Understanding t-mobile's zero-rating implementation. In *Proc. of SIGCOMM Workshop on Internet QoE*, 2016.

[19] A. M. Kakhki, A. Razaghpanah, A. Li, H. Koo, R. Golani, D. R. Choffnes, P. Gill, and A. Mislove. Identifying traffic differentiation in mobile networks. In *Proc. of IMC*, 2015.

[20] J. Kastrenakes. FCC fines verizon $1.35 million over 'supercookie' tracking. https://www.theverge.com/2016/3/7/11173010/verizon-supercookie-fine-1-3-million-fcc, March 2016.

[21] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson. Netalyzr: Illuminating the edge network. In *Proc. of IMC*, 2010.

[22] F. Li, A. M. Kakhki, D. Choffnes, P. Gill, and A. Mislove. Classifiers unclassified: An efficient approach to revealing IP-traffic classification rules. In *Proc. of IMC*, 2016.

[23] R. Mahajan, M. Zhang, L. Poole, and V. Pai. Uncovering performance differences among backbone ISPs with Netdiff. In *Proc. of USENIX NSDI*, 2008.

[24] J. Mayer. How verizon's advertising header works. http://webpolicy.org/2014/10/24/how-verizons-advertising-header-works/.

[25] Measurement Lab Consortium. ISP interconnection and its impact on consumer internet performance. http://www.measurementlab.net/blog/2014_interconnection_report, October 2014.

[26] D. Rayburn. Cogent now admits they slowed down netflix's traffic, creating a fast lane & slow lane. http://blog.streamingmedia.com/2014/11/cogent-now-admits-slowed-netflixs-traffic-creating-fast-lane-slow-lane.html, November 2014.

[27] P. Rodriguez and V. Fridman. Performance of PEPs in cellular wireless networks. In *Web content caching and distribution*, pages 19–38. Springer, 2004.

[28] F. Sarkar. Prevention of bandwidth abuse of a communications system, Jan. 9 2014. US Patent App. 14/025,213.

[29] P. Svensson. Comcast blocks some internet traffic. http://www.washingtonpost.com/wp-dyn/content/article/2007/10/19/AR2007101900842.html, October 2007.

[30] Switzerland network testing tool. https://www.eff.org/pages/switzerland-network-testing-tool.

[31] M. B. Tariq, M. Motiwala, N. Feamster, and M. Ammar. Detecting network neutrality violations with causal inference. In *CoNEXT*, 2009.

[32] N. Vallina-Rodriguez, S. Sundaresan, C. Kreibich, N. Weaver, and V. Paxson. Beyond the radio: Illuminating the higher layers of mobile networks. In *Proc. of MobiSys*, 2015.

[33] N. Weaver, C. Kreibich, M. Dam, and V. Paxson. Here Be Web Proxies. In *Proc. PAM*, 2014.

[34] N. Weaver, C. Kreibich, M. Dam, and V. Paxson. Here be Web proxies. In *Passive and Active Measurement (PAM)*, 2014.

[35] N. Weaver, R. Sommer, and V. Paxson. Detecting forged TCP reset packets. In *Proc. of NDSS*, 2009.

[36] X. Xu, Y. Jiang, T. Flach, E. Katz-Bassett, D. Choffnes, and R. Govindan. Investigating transparent web proxies in cellular networks. In *Proc. PAM*, 2015.

[37] Y. Zhang, Z. M. Mao, and M. Zhang. Detecting Traffic Differentiation in Backbone ISPs with NetPolice. In *Proc. of IMC*, 2009.

[38] Z. Zhang, O. Mara, and K. Argyraki. Network neutrality inference. In *Proc. of ACM SIGCOMM*, 2014.