OpenTD: Open Traffic Differentiation in a Post-Neutral World

Vijay Sivaraman[1], Sharat Chandra Madanapalli[1], Himal Kumar[2], and Hassan Habibi Gharakheili[1]
UNSW Sydney[1], Canopus Networks[2]

ABSTRACT

In the absence of network neutrality, consumers are vulnerable to arbitrary traffic discrimination policies applied by Internet Service Providers (ISPs). In this paper we propose a framework that gives ISPs flexibility to practice differentiation, while being open so consumers can make informed choices, and accountable so regulators can oversee adherence. We begin by outlining the SDN-based architecture of our solution, comprising the segregation of traffic into a chosen number of classes, and dynamic partitioning of bandwidth amongst classes based on utility functions. We then highlight the flexibility of our framework in accommodating a wide range of behaviors, from fully-neutral to per-application-type and per-subscriber-tier differentiation. We evaluate our scheme via simulations of real traffic mixes to show how ISP differentiation policies can be tuned to meet a range of user needs, and implement our scheme in a testbed network to demonstrate practical feasibility. We believe our proposal is a promising approach to keeping ISPs, consumers, and regulators happy in a post-neutral world.

1 INTRODUCTION

Network neutrality – the principle that all packets in a network should be treated similarly irrespective of their source or content – has been vigorously debated in the public domain for nearly 20 years [11]. The pendulum has swung back-and-forth several times in the US, with the FCC mandating neutrality in broadband networks in 2015, and then repealing it in 2017. The US has the underlying problem of lack of competition in fixed-line broadband, with a majority of homes having a choice between only one telecom network and one cable network [6], which creates public pressure to regulate the monopolistic ISPs (interestingly, mobile networks in the US have seen more competition and been largely exempt from the net-neutrality debates). Unlike the US, several countries in the world have encouraged competition in broadband service, and in some cases even paid for national broadband infrastructures from the public purse (e.g. Singapore, Australia, New Zealand, Korea, Japan), which creates a competitive marketplace of ISPs – for example, a subscriber in Australia can today choose from tens of ISPs over the NBN [7]. In the presence of such healthy competition, we believe it is unnecessary to impose neutrality on all ISPs; instead, they can be allowed (even encouraged) to differentiate their service in unique ways, and the market can decide how much their offering is worth (and indeed, if a net-neutral ISP dominates, so be it).

The absence of neutrality creates suspicion that ISPs will abuse their power and impose arbitrary traffic discrimination policies without consumer knowledge or consent. The research challenge therefore is to develop a traffic discrimination framework that is flexible enough to allow ISPs to innovate and differentiate their offerings, while being open enough for consumers to compare the offerings, and accountable enough for regulators to validate the actions of the ISPs. We believe such a framework has not been proposed in the literature to-date.

Our framework attempts to meet the requirements of the various stakeholders in the following way: For **ISPs**, our framework gives *flexibility* to specify differentiation policies based on any attribute(s), such as content type, content providers, subscriber tier, or any combination thereof. For example, the framework allows prioritizing streaming video over downloads, giving gold subscribers a greater share of bandwidth than bronze ones, or even restricting certain applications or content. Needless to say, the framework's theoretical flexibility will in practice be constrained by the legal and regulatory environment of the region, and ultimately by market forces. For **consumers**, our framework is *open* and allows them to see and compare the policies on offer from the various ISPs, in terms of the number of traffic classes each ISP supports, how traffic streams map to classes, and how bandwidth is shared amongst classes at various levels of

SOSR '19, April 3–4, 2019, San Jose, CA, USA

ACM ISBN 978-1-4503-6710-3/19/04…$15.00
https://doi.org/10.1145/3314148.3314354

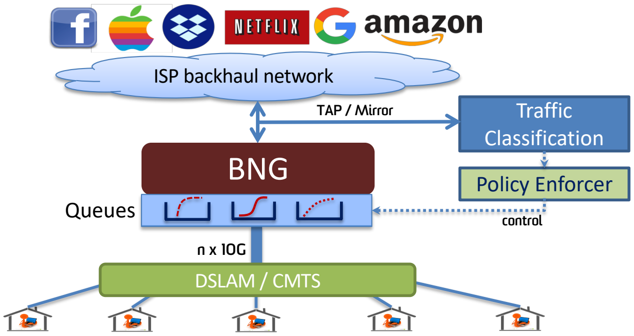


Figure 1: System architecture

congestion. This allows consumers to identify ISPs that better support their specific tastes, be it gaming or streaming video or large downloads, or indeed non-discrimination. Further, in exposing its policy, the ISP does not reveal any sensitive information about their network (such as provisioned bandwidth) or their subscriber base (such as numbers in each tier). Lastly, for **regulators**, our framework provides some level of *accountability* so that the differentiation behavior during congestion is verifiable as being conformant with the ISP's stated discrimination policies. We acknowledge that this paper represents only the first step towards a fresh solution to the inherently complex problem of network neutrality, and further work will be needed to address each of the issues in greater detail.

Our specific contributions are as follows: We begin by outlining our framework in terms of its architecture and rationale in §2. We then discuss its flexibility in accommodating a wide range of intended behaviors in §3. In §4 we conduct simulations using real traffic mix taken from a Tier-1 carrier to demonstrate how framework parameters can be tuned to achieve improved user experience. Finally, we implement our framework using SDN in a test-bed environment in §5. Relevant prior work is summarized in §6 and the paper concluded in §7.

2 ARCHITECTURE OF OPENTD

The focus of OpenTD is on congested links, such as at the local-exchange/central-office where traffic to/from subscribers (typically thousands in number) on the broadband access network (based on DSL, cable, or fiber) is aggregated by one or more broadband network gateways (BNGs), as shown in Fig. 1. Congestion is most prominent here, since the ISP will invariably oversubscribe the capacity available at the BNG. For example, if 5,000 subscribers in an access network aggregated at a BNG are each offered a 20 Mbps plan, the ISP would not provision 100 Gbps of backhaul capacity on the BNG, since that would be excessive in cost (for example, the list price of bandwidth on the Australian NBN [5] shows that 10 Gbps capacity costs $175,000 per-month). The ISP would therefore rely on statistical multiplexing to provision say a tenth of the theoretical maximum in order to save cost, equating to an aggregate bandwidth of 10 Gbps (or 2 Mbps per-user on average). Needless to say, this can cause severe congestion during peak hour when many users are active on their broadband connections. We now describe the elements of our framework that allow the ISP to deal with this congestion in an open, flexible, and accountable manner.

2.1 Per-Class Queueing and Flow Mapping

We begin by noting that OpenTD deals only with bandwidth management, and does not do path selection or latency control. Our framework therefore requires the ISP to specify the number of traffic classes (queues) they support at the congestion point, and how traffic streams get mapped to their respective classes. On the one extreme the ISP may have only one (FIFO) class, in which case they are net-neutral. On the other extreme they may have a class per-user per-application stream, akin to the IntServ proposal [3] – though theoretically permissible, this will require hundreds of thousands of queues making it infeasible in practice. A pragmatic approach would be for the ISP to support a small number (say 2 to 16) of classes – while this may sound somewhat similar to DiffServ [2], we emphasize that the number of classes, the mapping of traffic streams to classes, and their relative bandwidth shares are decided by the ISP and not mandated by any standard. For example, the ISP may choose to have three classes – one each for browsing, video, and large download streams.

The ISP has to clearly specify the criteria by which traffic flows get mapped to classes. For example, the ISP could state that flows that transfer no more than 4 MB each (aka *mice*) get mapped to the "browsing" class, flows that carry streaming video (deduced based on address prefixes, deep packet inspection, statistical profile measurement, or any other technique) map to the "video" class, and non-video flows that carry significant volume (aka *elephants*) get mapped to the "downloads" class. Additional classes can be introduced as and when necessary, for example to have a separate class for video from specific providers such as Netflix; however, such changes need to be openly announced by the ISP, including the mapping criteria and bandwidth shares, as described next.

2.2 Bandwidth Sharing Amongst Classes

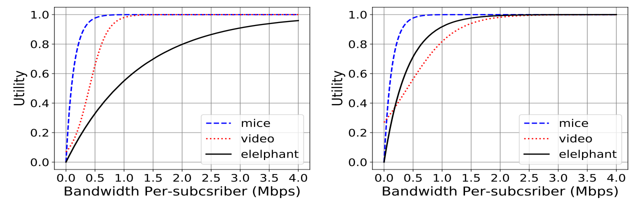
The framework has to specify the bandwidth sharing amongst classes in a way that: (a) is highly flexible so ISPs can customize their offerings as they see fit; (b) is mathematically rigorous and computable across the entire range of traffic conditions; (c) is simple to implement at high speeds; (d) does not require ISPs to reveal sensitive information including link speeds and subscriber counts; and (e) provides some level of accountability to regulators.

We rejected several possible approaches, including simplistic ones that specify a minimum bandwidth share per-class (as it may be variable with total capacity and is ambiguous when some classes do not offer sufficient demand) and complex ones (like in IntServ/DiffServ) requiring sophisticated schedulers. Instead, we choose the utility function framework that has been used successfully in economic theory to optimally partition a resource. The ISP is required to associate each class with a utility function that captures the "value" of bandwidth to that class. Though utility functions have been discussed in the networking literature for well over two decades [17], they usually start with the bandwidth "needs" of an application (voice, video or download) stream, and attempt to distribute bandwidth resources to maximally satisfy application needs. By contrast, our framework flips the viewpoint by having the ISP determine the utility function for a class, based on their perceived value of that traffic class in their network. Stated differently, a class' utility function is a way for the ISP to state (openly) how much they value that class at various levels of resourcing. As we will show soon, this gives ISPs high flexibility to customize their differentiation policy, while consumers and regulators benefit from the open knowledge that they can use to compare ISPs.

3 BANDWIDTH SHARING IN OPENTD

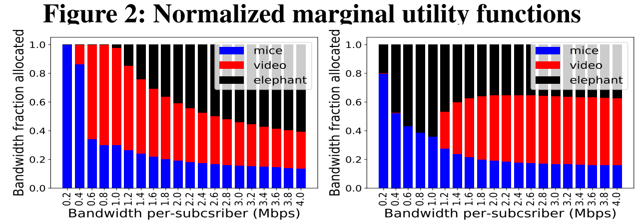
Bandwidth on the congested link is shared amongst classes in a way that maximizes global utility. Stated mathematically, let d_i denote the traffic demand of class-i, and $U_i(x_i)$ its utility when allocated bandwidth x_i. For given capacity C, the objective then is to determine x_i that maximizes $\sum_i U_i(x_i)$, where $\sum_i x_i = C$ and $\forall i : x_i \leq d_i$. We use algorithms for determining this numerically that are well-known in the literature [17] – a simple approach to compute optimal allocations is to take the derivate of the utility function $\partial U_i/\partial x_i$ (also known as the marginal utility function), and distribute bandwidth amongst the classes such that their marginal utilities are equalized. In practice, this optimization would be run periodically (say every second), and the per-class demand for an epoch would be set to $(1 + \epsilon)$ times the offered load of that class from the previous epoch, where ϵ is a small number that allows the class to "expand" from epoch to epoch using closed-loop congestion control mechanisms employed by the flows therein.

Let us now look at some examples of how an ISP can select utility functions. In example 1, the ISP wants to implement a pure priority system wherein class-i gets priority over class-j. The ISP can then select a linear utility function $U_i(x_i) = a_i x_i$ for class-i. This ensures that the marginal utility $\partial U/\partial x$ is always higher for class-i than class-j when $a_i > a_j$, and class-i's bandwidth demand is therefore always met before class-j receives any allocation. In example 2, the ISP wants to divide bandwidth amongst the classes in a given proportion: for example, browsing gets 30% of bandwidth, video 50%, and downloads 20%. Then the ISP can choose utility functions



(a) Video-friendly ISP-1 (b) Download-friendly ISP-2

Figure 2: Normalized marginal utility functions



(a) Video-friendly ISP-1 (b) Download-friendly ISP-2

Figure 3: Bandwidth share per class

of the form $U_i(x_i) = \sqrt{a_i x_i}$, which ensures that the marginal utilities of the classes are equalized when a_i/x_i is the same for each class, namely when bandwidth for class-i is proportional to a_i.

The flexibility of the utility function framework allows it to accommodate a much wider variety of bandwidth distribution schemes than the simple ones illustrated above. Considering again a three class scenario, we show in Fig. 2(a) an example of a "video-friendly" ISP-1 that uses utility functions U_m, U_v, U_e for mice, video, and elephants as follows:

$$U_m = 1 - e^{-1.5x}; U_v = 1/(1 + e^{-1.3(x-2.0)}); U_e = 1 - e^{0.16x} \quad (1)$$

and in Fig. 2(b) a "download-friendly" ISP-2 that uses the following utility functions for mice, video, and elephants respectively:

$$U_m = 1 - e^{-1.5x}; U_v = 1/(1 + e^{-0.5(x-2.0)}); U_e = 1 - e^{0.50x} \quad (2)$$

Note that in order to keep information on provisioned bandwidths (both aggregate and per-consumer) private, the utility curves released by the ISPs above are scaled versions, whereby the x-axis denotes the provisioned backhaul capacity divided by the number of subscribers multiplexed on that link.

Fig. 2 shows that ISP-1 values video more at low bandwidths than ISP-2, while ISP-2 conversely values downloads more than video at low bandwidths. At higher bandwidth (4 Mbps per-subscriber and above), the differences in utility become less significant. This is indeed borne out when we compute the bandwidth allocation as a function of provisioned bandwidth per-subscriber, as shown in Fig. 3, when each class offers sufficient demand. Fig. 3(a) shows that ISP-1 will prioritize video over downloads if the bandwidth provisioned per-subscriber is 1.0 Mbps or lower, while ISP-2 will conversely prioritize downloads over video over this range as shown in Fig. 3(b). However, as the provisioned bandwidth per-customer increases, we note that the allocation starts becoming more balanced across the classes for both ISPs. As noted earlier, our framework does not require the ISP to reveal the per-subscriber bandwidth at the aggregation point, as this is commercially sensitive information. Also, the average bandwidth provisioned per-user of 2-4 Mbps is in the ballpark of numbers we are aware from ISPs, as they rely on statistical multiplexing whereby only a fraction of users are active at any point in time. Further, the same utility functions can be applied at any link in the ISP network by scaling them to the total bandwidth provisioned on that link.

Verifying that the ISP is honestly applying the bandwidth partitioning is not easy – indeed, even the special case of whether or not a network is doing any discrimination at all is in itself very challenging to determine without inside knowledge [19]. We suggest an approach whereby the ISP is required to submit to the auditor a profile of the arriving and serviced traffic volumes pertaining to each of their traffic classes over a given time period. The auditor can make a determination on whether the per-class arriving traffic is reasonable (e.g. based on known patterns), and verify that the bandwidth allocations are conformant with utility maximization as per the advertised utility curves, thereby providing some measure of assurance to the regulator that the ISP is adhering to their stated discrimination policy.

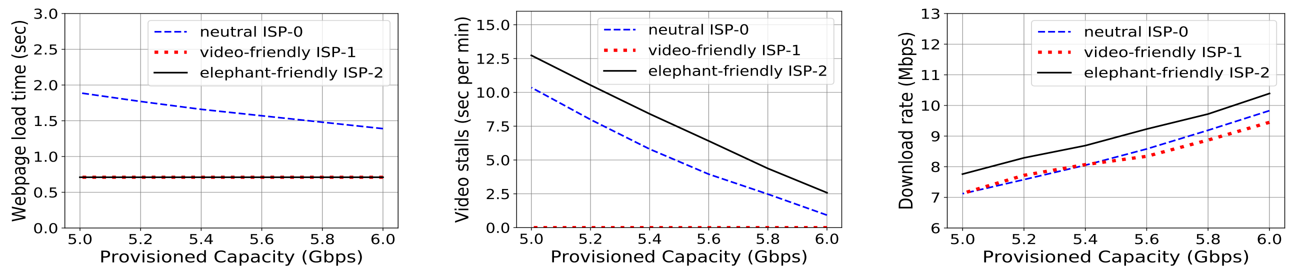
4 MEASURING USER EXPERIENCE

We built an idealized simulator to evaluate the impact of the differentiation framework on user experience. A single link at the BNG that aggregates multiple subscribers over the access network is considered, wherein each traffic flow is classified into one of multiple queues, and bandwidth is partitioned amongst the classes based on their respective utility functions. Traffic is modeled as a fluid, and the simulation progresses in time slots. In each time slot, each active flow submits its request (number of bits it wants transfered in that slot) to the link scheduler; the requests are aggregated by the scheduler into classes, allocations made to each class in a way that maximizes overall utility for the given demands (as per algorithm described at the beginning of §3), and the bandwidth allocated to each class is shared fairly amongst active flows in that class. Each flow implements TCP dynamics to adjust its request for the subsequent time slot based on the allocation in the current slot – if the request is fully met, it increases its rate (linearly or exponentially depending on whether it is in congestion-avoidance or slow-start phase), while if the request is not fully met it reduces its rate (by half under moderate congestion and to one MSS-per-RTT under severe congestion, determined by whether the allocation is at least half of its request or not). Further, the rate of any flow is limited by its access link capacity. While our fluid simulation model does not fully capture all the packet dynamics and variants of TCP, we believes it captures its essence, and allows us to simulate large workloads quickly with reasonable accuracy.

The simulation parameters are adjusted as follows: the access links have capacity uniformly distributed in [10,30] Mbps, and multiplex at a link whose capacity we provision in the range [5, 6] Gbps. The simulation slot size is set to 100 μsec, TCP MSS to 1500 bytes and RTT uniformly in the range [50,250] msec. We simulate traffic representative of 3000 subscribers, comprising: browsing flows arriving at 200 flows/sec and loading a web-page exponentially distributed in size with mean 1 MB; elephant flows arriving at 4 flows/sec with exponentially distributed download volume of mean 100 MB; and video flows arriving at 4 flows/sec at HD quality, with playback rate of 5 Mbps and playback buffer replenished by an underlying TCP process; further, the playback buffer holds up to 30 seconds of video, is replenished when occupancy falls below 10 seconds worth, and playback starts as soon as 2 seconds worth of video is ready in the buffer. We believe this simulated behavior of video streams is simplistic but captures the dynamics of real streaming video from providers such as Youtube and Netflix to a reasonable degree of approximation. These settings yield a traffic mix of about 28% browsing, 38% video, and 34% downloads – this is reasonably consistent with the mix we have observed in operational networks.

User experience is measured in terms of three aspects: page-load time aka average flow completion time (AFCT) in seconds for browsing flows; playback stalls (in seconds per minute) for streaming video flows; and mean rate (in Mbps) for elephant/download flows. These are displayed continuously by the simulation process via the user interface. The base case for our simulation is a net-neutral ISP-0 who has only a single traffic class, and provisions bandwidth in the range 5-6 Gbps to serve the 3000 subscribers. This is compared to a video-friendly ISP-1 who uses utility functions: $U_m(x_m) = \sqrt{0.4x_m}$, $U_v(x_v) = \sqrt{0.5x_v}$ and $U_(x_e) = \sqrt{0.1x_e}$ for mice, video, and elephant classes respectively, in essence assigning them bandwidth in the ratio 4:5:1, and a download-friendly ISP-2 who uses utility functions $U_m(x_m) = \sqrt{0.4x_m}$, $U_v(x_v) = \sqrt{0.3x_v}$ and $U_(x_e) = \sqrt{0.3x_e}$ yielding a bandwidth ratio of 4:3:3.

Fig. 4 depicts the user-experience across the three ISPs as a function of the provisioned bandwidth. We note from Fig. 4(a) that the web-page load time is lower at 0.71 sec with ISP-1 and ISP-2, unlike in neutral ISP-0 where mice flows intermix with video and downloads to inflate load times to 1.39-1.89 seconds. Video traffic experiences stalls of 0.92-10.36 seconds on average with ISP-0, as shown in Fig. 4(b), whereas ISP-1 eliminates stalls by virtue of giving higher utility to the video class, and ISP-2 degrades video by allowing stalls of 2.58-12.73 seconds per video play. Conversely, download rates are higher in the download-friendly ISP-2 (7.76-10.39 Mbps) and lower in the video-friendly ISP-1 (7.13-9.45 Mbps)



(a) Web-page load time (sec) (b) Video stalls (seconds per minute) (c) Download rate (Mbps)

Figure 4: User experience across neutral, video-friendly, and download-friendly ISPs

compared to the neutral ISP-0 (7.12-9.83 Mbps), as shown in Fig. 4(c). This confirms that the ISP's publicly stated utility functions are corroborated in the resulting user experience, and our framework empowers ISPs to adjust their class utility functions to differentiate their offerings in the market.

5 IMPLEMENTING OPENTD

We implemented the OpenTD framework in an SDN testbed depicted in Fig. 5. The objectives of this experimental setup are to demonstrate the feasibility of our scheme with real equipment and traffic, and to evaluate the efficacy and flexibility of traffic differentiation for real video, web browsing, and large transfers.

For our BNG, we use a NoviFlow SDN switch that is controlled by a Ryu SDN controller and connects the subscribers to the Internet via our campus network. The switch has 16 x 10G ports, supports millions of flow entries, thousands of flow modifications per second, and thousands of queues, and therefore is very capable of supporting a large number of traffic classes, and dynamically mapping flows to classes and adjusting class bandwidths at high traffic rates. In our lab we configure it to operate as a 100 Mbps BNG. We use three standard machines running Ubuntu 16.04, each represents a broadband subscriber – A, B, and C. We have developed a traffic generator tool (written in Python) installed on each machine. Three classes of traffic namely mice, video, and elephant are considered: mice flows are generated by fetching a set of webpages using `requests` library in Python; elephant flows are generated using `wget` Unix downloader tool; and video flows are generated by playing YouTube and Netflix videos in a

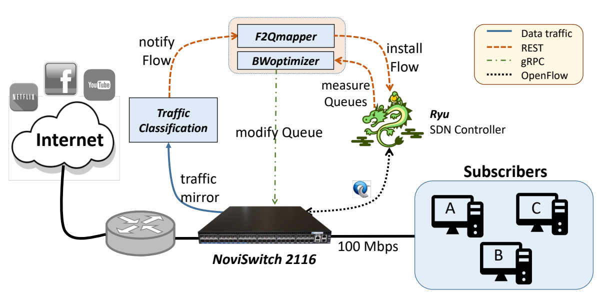


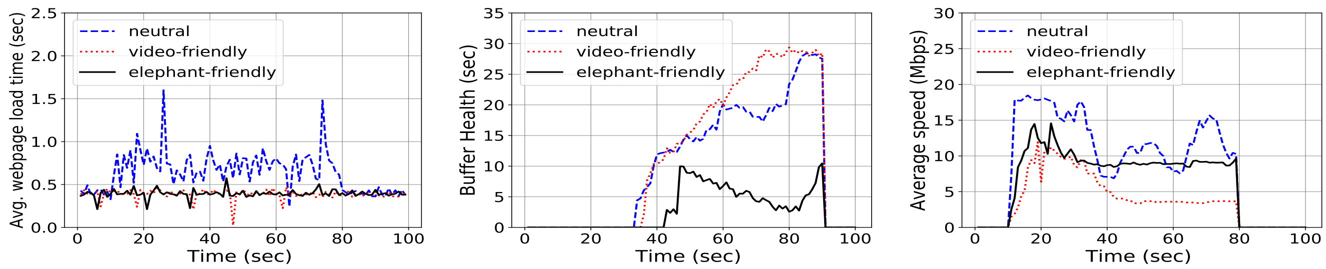
Figure 5: OpenTD implementation

Chrome browser automated using Python `Selenium` library. The module also collects performance metrics (i.e. web-page load time for mice, buffer health and stalls for videos, download rates for elephants) of traffic streams running on each client machine. Flows associated with each class are aggregated using the OpenFlow group entry on the SDN switch – each group is mapped to a corresponding queue.

Additionally, we have written three programs all in Golang: Traffic Classification is an application that identifies the class of a traffic flow in real-time – it outputs its 5-tuple and class; F2Qmapper makes a REST call to the Ryu controller, mapping the identified flow to its appropriate queue (via group entry); BWoptimizer periodically computes the max rate of queues according to utility curves given the real-time measurement of demand in each queue (class), and modifies queues rate using a gRPC call. Our SDN switch only allows us to modify the queues rate at a step of 10 Mbps. We therefore employed a simple utility curve with square root function (i.e. $\mathrm{U_i}(x_i) = \sqrt{c_i x_i}$), which yield allocation proportional to c_i for class-i.

We run three scenarios of experiments namely neutral ISP, video-friendly ISP, and elephant-friendly ISP – each run lasts for 100 seconds. In all experiments, traffic is generated in a way that machines A, B, and C respectively emulate browsing-heavy, download-heavy and video-heavy subscribers. At time 1s, mice flows begin on A. At 10s, the machine B starts four downloads (that run concurrently until 80s). The traffic mix remains elephant and mice until 30s when the machine C plays a couple of 4K videos on Youtube until 90s. Fig.6 depicts the average performance metric for each class (of subscriber). A neutral ISP-0 imposes no differentiation to the traffic. A video-friendly ISP-1 allocates the capacity to mice, video and elephant classes in the ratio of 3:5:2 respectively. Lastly, an elephant-friendly ISP-2 allocates bandwidth to these classes in the ratio of 3:2:5.

It is seen in Fig. 6(a) that the web-page load time is the worst in a neutral scenario (shown by dashed blue lines). This is because of the high demand from both video and elephant flows that aggressively consume the link bandwidth. Instead,



(a) Mice (b) Video (c) Elephant

Figure 6: Experiment results(mean): (a) mice page load time, (b) video buffer health, and (c) elephant download rate

both video-friendly and elephant-friendly ISPs offer a consistent browsing experience with 50% reduction in the average load time compared to the neutral ISP, since 30% of the total capacity is provisioned to mice flows during congestion.

We illustrate the performance of video flows in Fig. 6(b). In the neutral scenario, videos get affected by the heavy load from elephants and are unable to reach the peak buffer capacity until elephants stop at 80s. The video-friendly ISP, on the other hand, ensures that videos get good experience by limiting the downloads during congestion period. The video experience on an elephant-friendly network would not be great as expected – nevertheless, the increase in buffer capacity is observed after the downloads have stopped.

Lastly, elephants perform the best in the neutral scenario causing mice and videos to suffer, as shown in Fig. 6(c). The download speed fluctuates significantly on arrival of videos. Downloads on the elephant-friendly network hit the peak rate of 16Mbps and come down to about 9 Mbps after videos begin, while giving some room to mice flows too. In the video-friendly scenario, the rate of downloads falls slightly compared to the elephant-friendly at the beginning, but it is suppressed heavily as soon as videos arrive. Additional results can be found in Appendix.

6 PRIOR WORK

Net neutrality has been heavily debated over the past several years [1, 8, 14, 19] among policy makers, researchers, and activists covering its economic, technological, and societal aspects. Work in [14] developed an analytical model for monopoly/duopoly Internet ecosystems, and used game-theory analysis to derive the Nash equilibrium between ISPs and content providers under a paid privatization condition. In [8], a model for two-sided Internet pricing is developed to assess the effect of a non-neutral market where ISPs are allowed to charge content providers for access to consumers. Work in [19] highlights the need for measuring the neutrality of networks, and surveys existing techniques (and their challenges) for detecting traffic differentiation on the Internet. For the measurement of net neutrality, the IETF has developed an Internet draft [15] based on the new European open Internet regulation.

Net neutrality regulations around the globe are surveyed in our prior work [11]. As mentioned earlier, the FCC in 2017 repealed [20] net neutrality regulations established in 2015. In a post-neutral world, it is conceivable for ISPs to prioritize (i.e. offer faster lanes to) certain classes of traffic without legal reactions, as long as they disclose it to the public on their own websites or to the FCC [16], though no formal model for disclosure has emerged.

In terms of differentiation techniques, there is a body of literature on adopting utility theory-based algorithms for allocation of network resources [4, 12, 13, 17, 21]. In a most recent work [13], the authors employ utility functions specific to network attributes such as bandwidth, delay, and jitter for representative services including voice, video, and data to select the best interface in heterogeneous wireless networks. Network selection aims to maximize the users' perceived quality for individual applications. Lastly, there are a number of proposals [9, 10, 18] that use SDN to manage the quality of service for various applications on the network. Our recent work in [10] considers a three-party approach for dynamic provisioning of Internet fast-lanes on consumers broadband link. In this approach, fast-lanes are dynamically invoked via open APIs available for any content provider to invoke for a specific traffic stream.

7 CONCLUSION

In this paper we have proposed a framework called OpenTD that fills the vacuum left by the demise of network neutrality. OpenTD allows differentiation, but in a flexible and open way so ISPs can be explicit about their policies, consumers can compare ISPs, and regulators can oversee conformance. We developed an architecture that is flexible and scalable, and an algorithm for dynamic bandwidth management, while preserving ISP's private information on subscriber numbers and provisioned capacity. We simulated our scheme with realistic traffic mixes to demonstrate how ISPs can differentiate their services in terms of user experience, and built a fully functional prototype using SDN. We believe our framework is a worthwhile starting point for how the community shapes the post-neutral world.

Figure 7: Performance for Youtube (top left), Netflix (top right), browsing (bottom left), downloads (bottom right)

REFERENCES

[1] E. Altman, A. Legout, and Y. Xu. 2010. Network Non-neutrality Debate: An Economic Analysis. *ArXiv e-prints* (Dec. 2010). arXiv:cs.NI/1012.5862

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. 1998. *An Architecture for Differentiated Services*. RFC 2475. IETF. https://tools.ietf.org/html/rfc2475

[3] R. Braden, D. Clark, and S. Shenker. 1994. *Integrated Services in the Internet Architecture: an Overview*. RFC 1633. IETF. https://tools.ietf.org/html/rfc1633

[4] Li Chen, Bin Wang, X. Chen, Xin Zhang, and Dacheng Yang. 2011. Utility-based resource allocation for mixed traffic in wireless networks. In *Proc. IEEE INFOCOM*.

[5] NBN Co. 2018. NBN Price list. https://www.nbnco.com.au/content/dam/nbnco2/documents/sfaa-wba2-product-catalogue-price-list_20170724.pdf

[6] A. Cooper and I. Brown. 2015. Net Neutrality: Discrimination, Competition, and Innovation in the UK and US. *ACM Trans. Internet Technol.* 15, 1 (March 2015), 2:1–2:21.

[7] A. Daly. 2014. Net Neutrality in Australia: an emerging debate. *2nd Report of the UN IGF Dynamic Coalition on Network Neutrality* (Sept. 2014), 43–58.

[8] N. Economides and J. Tag. 2012. Network neutrality on the Internet: A two-sided market analysis. *Information Economics and Policy* 24, 2 (2012), 91 – 104.

[9] A. D. Ferguson, A. Guha, C. Liang, R. Fonseca, and S. Krishnamurthi. 2013. Participatory Networking: An API for Application Control of SDNs. In *Proc. ACM SIGCOMM*. Hong Kong, China.

[10] H. Habibi Gharakheili, V. Sivaraman, T. Moors, A. Vishwanath, J. Matthews, and C. Russell. 2017. Enabling Fast and Slow Lanes for Content Providers Using Software Defined Networking. *IEEE/ACM Transactions on Networking* 25, 3 (June 2017), 1373–1385.

[11] H. Habibi Gharakheili, A. Vishwanath, and V. Sivaraman. 2016. Perspectives on Net Neutrality and Internet Fast-Lanes. *SIGCOMM Comput. Commun. Rev.* 46, 1 (Jan. 2016), 64–69.

[12] M. Katoozian, K. Navaie, and H. Yanikomeroglu. 2009. Utility-based adaptive radio resource allocation in OFDM wireless networks with traffic prioritization. *IEEE Transactions on Wireless Communications* 8, 1 (Jan 2009), 66–71.

[13] G. Liang and H. Yu. 2018. Network selection algorithm for heterogeneous wireless networks based on service characteristics and user preferences. *EURASIP Journal on Wireless Communications and Networking* 2018, 1 (17 Oct 2018), 241.

[14] R. T. B. Ma and V. Misra. 2013. The Public Option: A Nonregulatory Alternative to Network Neutrality. *IEEE/ACM Transactions on Networking* 21, 6 (Dec 2013), 1866–1879.

[15] K. Nieminen. 2017. *Net Neutrality Measurements: Regulatory Use Case and Problem Statement*. Internet-Draft draft-ietf-opsawg-mud-18. IETF Secretariat. https://tools.ietf.org/id/draft-nieminen-ippm-nn-measurements-01.txt

[16] The Washington Post. 2018. The FCC's net neutrality rules are officially repealed today. Here's what that really means. https://goo.gl/46oHZG

[17] R. Rajkumar, C. Lee, J. Lehoczky, and D. Siewiorek. 1997. A resource allocation model for QoS management. In *Proc. Real-Time Systems Symposium*. San Francisco, CA, USA.

[18] M. Said Seddiki, Muhammad Shahbaz, Sean Donovan, Sarthak Grover, Miseon Park, Nick Feamster, and Ye-Qiong Song. 2014. FlowQoS: QoS for the Rest of Us. In *Proc. ACM HotSDN*. Chicago, Illinois, USA.

[19] T. Garrett and L. E. Setenareski and L. M. Peres and L. C. E. Bona and E. P. Duarte. 2018. Monitoring Network Neutrality: A Survey on Traffic Differentiation Detection. *IEEE Communications Surveys Tutorials* 20, 3 (Q3 2018), 2486–2517.

[20] The New York Times. 2018. F.C.C. Repeals Net Neutrality Rules. https://goo.gl/9LxTxD

[21] L. Wang and G. G. S. Kuo. 2013. Mathematical Modeling for Network Selection in Heterogeneous Wireless Networks – A Tutorial. *IEEE Communications Surveys Tutorials* 15, 1 (Q1 2013), 271–292.

ACKNOWLEDGMENTS

The authors thank the anonymous reviews who gave us valuable feedback on our submission, and a special thanks to our shepherd, Marco Canini, for his help and suggestions during the revision process.

APPENDIX

We illustrate in Fig. 7 results from another set of experiments that show the ability of our framework to reduce aggregate bandwidth requirement while improving user experience. This figure captures the health of Youtube buffers (top left) and Netflix buffers (right top), as well as web-page load times (bottom left) and rate for large downloads (bottom right). The experiment is conducted in four phases - the first phase sets the baseline with aggregate provisioned bandwidth 100 Mbps and neutral behavior. In this phase, web-page loads are found to average 0.8 seconds, Youtube 4k video takes 25 seconds to fill its buffers, Netflix plays at 480p resolution and takes 60 seconds to fill its buffers, while downloads average 60 Mbps.

In the second phase, the aggregate provisioned bandwidth is reduced by 20%, namely to 80 Mbps. The user experience degrades as one would expect: web-pages take 1.1 seconds to load on average, Youtube takes 80 seconds to fill its buffer, Netflix takes 75 seconds, and downloads get 40 Mbps.

With bandwidth kept at 80 Mbps, the third phase of the experiment enables the OpenTD solution with utility curves tuned to achieve weighted priorities in the ratio 25:50:25 for browsing, video, and downloads respectively. It is now observed that web-page load times reduce to 0.34 seconds, the Youtube 4k streams takes 60 seconds to fill buffers while the Netflix stream is now able to operate at 720p and takes only 10 seconds to fill its buffers – these performance improvements come at the cost of reducing average download speeds to 20 Mbps. For the last phase of our experiment we configure the utility functions to realize priority for video over browsing over downloads. In this phase, web-page load times average 0.38 seconds, Youtube and Netflix take only 10 and 5 seconds respectively to fill buffers, and downloads get throttled to 15 Mbps. These experiments confirm that OpenTD can be tuned to save as much as 20% in bandwidth costs while preserving (and even enhancing) user experience for browsing and video streams.