Ascertaining the Reality of Network Neutrality Violation in Backbone ISPs

Ying Zhang
University of Michigan

Z. Morley Mao
University of Michigan

Ming Zhang
Microsoft Research

ABSTRACT

On the Internet today, a growing number of QoS sensitive network applications exist, such as VoIP, imposing more stringent requirements on ISPs besides the basic reachability assurance. Thus, the demand on ISPs for Service Level Agreements (SLAs) with better guarantees is increasing. However, despite overprovisioning in core ISP networks, resource contention still exists leading to congestion and associated performance degradations. For example, residential broadband networks rate-limit or even block bandwidth intensive applications such as peer-to-peer file sharing thereby violating network neutrality. In addition, traffic associated with specific applications, such as Skype, could also be discriminated against for competitive business reasons.

So far, little work has been done regarding the existence of traffic discrimination inside the core of the Internet. Due to the technical challenges and widespread impact, it seems somewhat inconceivable that ISPs are performing such fine-grained discrimination based on the application content. Our study is the first to demonstrate evidence of network neutrality violations within backbone ISPs. We used a scalable and accurate monitoring system – NVLens – to detect traffic discrimination based on various factors such as application types, previous-hop, and next-hop ASes. We discuss the implication of such discrimination and how users can counter such unfair practices.

1 INTRODUCTION

The topic of network neutrality on today's Internet is a highly contentious one. Previously, users assumed ISP networks are neutral to carry traffic without any preferential treatment. Edge customers can instrument their own policies for traffic management by for example blocking certain traffic using firewalls at the edge of the Internet. So customers expected that ISPs would not treat traffic differently based on properties other than the basic information required for forwarding, *e.g.,* destination IP address. In violation of network neutrality, traffic properties suspected to be used to perform discrimination include application types inferred from port numbers or payload data, previous-hop network, and next-hop network.

Various residential broadband networks, such as Comcast, are known to be violating network neutrality, by restricting the bandwidth usage of peer-to-peer file sharing applications. Network neutrality has different technical definitions and feasibility in various types of network models [1, 2]. Several research proposals exist for counteracting discrimination relying on encryption and multipath routing [3, 4], along with ideas to block traffic via auctions under the bandwidth shortage [5]. Given the potential detrimental effect on traffic which can be given lower priority, it is critical for end-users to first *detect* which ISP is violating network neutrality and to understand the policies for discriminating against specific traffic types. Beverly *et al.* presented the first study of the port blocking behavior that violates neutrality [6]. Related to our work, POPI is a tool for determining the router forwarding policy via end host measurements [7], but it only focuses on preferential treatment based on port numbers. Their methodology of saturating the link with high traffic volume is unsuitable for backbones.

No detailed and comprehensive study on the current practice of traffic discrimination, particularly inside the core ISPs, currently exists. And yet, traffic differentiation in the core has a much wider scope of impact, as such policies affect much more traffic compared to policies near the edge of the Internet. Knowing which ISPs perform discrimination and how they perform it is a critical first step towards identifying alternatives to address the network neutrality issues.

Our work is the first to demonstrate concrete evidence of network neutrality violations in backbone ISPs and analyze the extent of their violations. We developed a scalable and accurate distributed measurement methodology called NVLens(*Neutrality Violation Lens*[1]) to monitor ISP's loss and delay behavior in order to identify traffic discrimination based on factors such as applications, previous-hop and next-hop ASes. Given the initial report on discrimination, we performed selective drill-down to deduce how discrimination is implemented.

Unlike ISP-centric SLA monitoring, which requires access to proprietary data, NVLens relies on minimal network cooperation and is entirely end system based, leading to easy deployment and accurate observation from the end system's perspectives. NVLens can be used as a simple tool by end users to detect network neutrality violation and similarly SLA compliance of any ISP. By studying 19 large ISPs covering major continents including North America, Europe, and Australia over several weeks, we discovered ISPs some-

[1]The common translation of "night vision lens" is also relevant here, as our monitoring continuously covers both day and night.

Type	Examples
Application types	packet header field (*e.g.,* src/dst port numbers, protocol type)
Application properties	data content, application protocol header (*e.g.,* HTTP header, IPSec header)
Network policies	routing info (previous-hop, next-hop AS, routing entry)
Traffic behavior	flow rate, packet size, flow duration, fragment bit
Available resources	router state (load, memory), time of day, location (*e.g.,* PoP)

Table 1: Information commonly used to determine policies for discrimination.

times do give different priority to traffic coming from different neighbors (previous-hop ASes). Discrimination based on the next-hop AS is less common. We also observed different priority for traffic associated with UDP and specific applications such as BitTorrent compared to HTTP traffic. The loss rate increase for discriminated traffic can be as high as 8% with up to 500ms increase in RTT.

2 NET NEUTRALITY VIOLATION

In this study, we define network neutrality as ISPs giving equal treatment to packets regardless of their application content, application types, and packet sources or destinations. Any differentiation behavior that violates network neutrality is called *discrimination*. Note that we broaden the previous definition [2] by not singling out customers who may receive better treatment. Therefore, the observed performance difference can result from distinct business contracts between provider and its customers. It is debatable that whether this type of discrimination should be considered as neutrality violation. In this work we also report such discrimination to enable different interpretations.

Packets contain plenty of information that an ISP can use to construct discrimination policies. Table 1 shows the potential factors used to determine the discrimination policy. First, an ISP may provide differentiated service depending on the application type for security or business reasons. Application types can be determined from transport layer protocol fields or application layer content information [8]. Even with encrypted traffic, such discrimination can be made using more sophisticated traffic flow information [9]. Second, an ISP can discriminate against traffic due to business relationships, based on their source/destinations or incoming/outgoing networks. This information can be easily gathered from packet headers and routing information. Third, an ISP can selectively enable discrimination depending on the resource conditions, *e.g.,* when resources are limited as indicated by high link utilization.

The feasibility of implementing packet discrimination in a backbone ISP network with many high-speed links is questionable due to the need to perform additional per packet processing. We discuss several techniques that an ISP can employ to implement relevant policies today.

Today's router already has support for various queuing mechanisms to fulfill the need of traffic engineering, ensuring quality of service and security guarantees. Figure 1 illustrates a common architecture for implementing the discrimination within an ISP. The ingress border routers perform traffic classification by marking packets according to priorities, which are determined by packet fields such as protocol, source, and destination. The marking usually occurs on the Type-of-Service (TOS) field in the IP header. The internal routers can carry out different queuing and dropping decisions according to the packet classification encoded within TOS by the border routers [10]. Different queuing mechanisms provide various services to traffic based on its priority, *e.g.,* priority queuing, proportional share scheduling, and policing [11]. These mechanisms differ in details of how and when the differentiation is carried out.

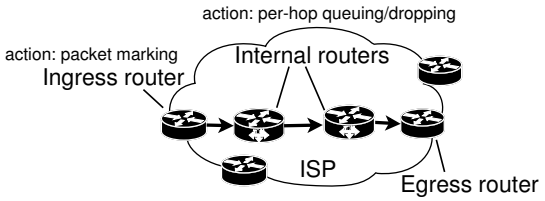

Figure 1: An example of discrimination implementation.

Besides router based mechanisms relying on packet header information, deep packet inspection (DPI) tools [12] allow ISPs to classify applications using packet content to understand application types. Although DPI devices are usually too expensive to be widely deployed, some current products claim to support up to 100 Gps links [13, 14] capable of searching for patterns in the payload using hardware support.

Given the feasibility of discrimination deployment, we studied all the factors shown in Table 1 except for the discrimination based on traffic behavior due to the limited resource of end-host based probing. This type of discrimination is also more difficult to implement by ISPs due to required per flow state information. NVLens enables us to discern which factor(s) may influence ISP's policies for preferential treatment of different classes of traffic. The design is extensible to other factors once they are known. The goal of detecting all these types of discrimination guides the methodology design of probing strategy and the probe packet composition in NVLens .

3 MEASUREMENT METHODOLOGY

This section describes the design of NVLens and illustrates how to monitor networks for neutrality compliance from end systems without any ISP cooperation. NVLens has the capability to detect three main types of network neutrality violations. Figure 2 illustrates the collaborative probing used to detect neutrality violations by a particular ISP based on factors such as application types and network policies (described in Table 1). Multiple ISPs were probed in parallel simultaneously to allow for accurate comparison. As shown in the figure, discrimination detection focuses on *ISP W* based on different traffic properties, *i.e.,* towards different next-hop ASes, from different previous-hop ASes, or based on differ-

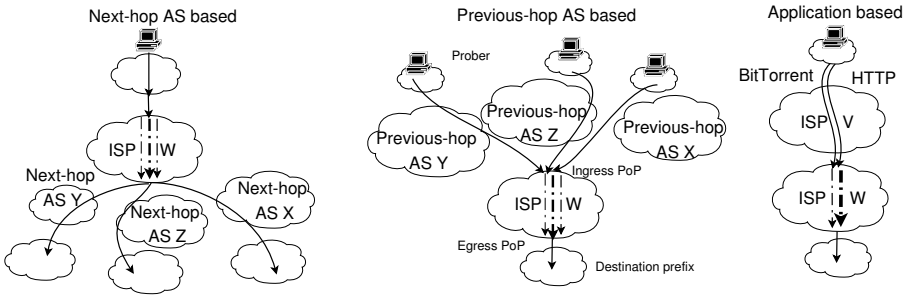

Figure 2: Collaborative probing to discover neutrality violations of different types.

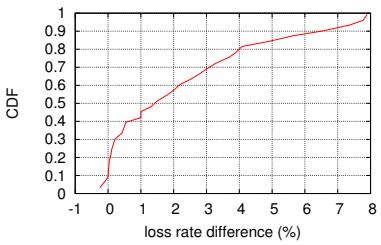

Figure 3: Loss rate difference between path pairs passing the test

ent application types.

Note that we focus on differences in performance metrics observed to identify traffic differentiation performed by the routers in ISPs. Many confounding factors could also cause differences in observed performance. First, different network paths in one ISP have different load leading to different performance observed. Even from one ingress to same egress, many equal-cost paths exist. Second, different application properties, *e.g.,* packet size, packet rate, can result in different performance measured. Third, external measurement artifacts, *e.g.,* heavily-loaded probing hosts, lossy reverse path, are also likely to create differences.

To rule out the impact of all these factors, we design our methodology carefully to eliminate the impact of most factors. For factors that are difficult to control, *e.g.,* impact of equal-cost paths, we use controlled experiments to confirm they would not introduce any systematic bias. In the following, we introduce our novel methodology to detect neutrality violation with low overhead.

3.1 Collaborative probing optimization

Probing overhead is always a concern in any active measurement study. For the purpose of discovering neutrality violation, it is particularly important to keep probing hosts lightly-loaded and to ensure short probing intervals. Otherwise, different performance might be caused by the heavily-loaded hosts or measurement conducted at different time periods. We use collaborative probing to ensure low probing overhead.

A typical backbone ISP consists of multiple PoPs (Points of Presence) at several geographic locations. In order to quantify the overall network neutrality compliance of an ISP and avoid the potential bias introduced by any particular path, NVLens should cover a reasonably large fraction of paths between distinct PoP pairs. Therefore, path selection strategy is a key issue for NVLens. Given a list of backbone ISPs, we couldn't afford to continuously probe all the destination prefixes on the Internet from all the probers. Instead, we devised an intelligent path selection strategy as follows for a probing interval: 1) Each three-tuple path (P_i, P_e, d) is traversed at least n times by probes from different probers; and 2) A prober does not conduct more than m probes. Here, s is a prober, d is a destination IP address, and P_i and P_e are the ingress and egress points in the target ISP respectively. Previous work [15] has shown this problem is an instance of the set covering/packing problem [16] for which we use a greedy algorithm as an approximation.

3.2 Loss rate and RTT measurement

NVLens measures both loss rate and roundtrip time (RTT) of a path which are simple performance metrics. To comply with the resource limits at each host, we take two steps to reduce probing overhead. First, NVLens only probes the hops that map to an ingress or an egress in one of the target ISPs instead of probing all the hops along a path. Since we are only interested in identifying ISP internal traffic discrimination between ingress-egress pairs, there is no need to probe other hops. Second, to measure the loss rate and RTT to a particular hop, NVLens sends probe packets with pre-computed TTL value which is expected to trigger ICMP time exceeded response from the corresponding router. In essence, the packet is similar to traceroute probes. However, since loss may occur in both directions, we use relatively large probe packets to increase the likelihood of inducing loss on forward paths only, which has been widely adopted in previous studies [17, 18]. NVLens probes each hop 200 times so that it can detect minimum loss rate of 0.5%. To reduce the chance of triggering ICMP rate limiting, NVLens probes each hop only at most once per second.

3.3 Application-specific probing

We use NVLens to explore how backbone ISPs preferentially treat various real-time and QoS sensitive applications. We choose five representative applications with distinct traffic characteristics in our study: UDP, HTTP, BitTorrent (P2P file sharing), Skype (VoIP), and World of Warcraft or WoW (online gaming). To avoid the overhead of comparing each pair of applications, we use HTTP traffic as the baseline. Since HTTP is the most widely-used Internet application, we assume it does not receive any preferential treatment, *i.e.,* representing the normal performance that most applications will experience.

The following steps are taken to eliminate the impact of all possible confounding factors we can think of. First, we classify applications into two groups: large packets of 200 bytes (HTTP, UDP, BitTorrent), small packets of 80 bytes (HTTP, Skype, World of Warcraft). This classification is based on empirical observation of corresponding applications, while observing the bandwidth constraints of probe hosts. We use controlled experiments to verify that most observed packet loss occurred on forward paths. We measure

three types of application at the same time, using the same probe hosts, for the same paths, with the same packet size.

To accurately represent the application behavior, we construct application-specific packets with the corresponding payload captured from real application traces. This is especially important for proprietary applications such as Skype or WoW whose protocols are not not publicly known. Given that data packets are often exchanged after a few control packets, we first transmit 9 to 25 small application-specific control packets at one packet per second rate. These packets help ISPs identify and potentially discriminate subsequent data packets. Control packets are identified using either known protocol specification (*e.g.,* for BitTorrent) or timing and packet size behavior (*e.g.,* for Skype), as there is usually a large gap between the exchange of control and data packets in both interpacket timing and packet size. Also note that control packets are constructed with sufficiently large TTLs, meaning all the routers along the path up to the last ISP's egress router can observe the control packets in case routers use such information to store state needed for traffic discrimination[2].

4 EXPERIMENTAL RESULTS

This section presents our experimental results that provide insights on how network neutrality is violated on today's Internet. To ensure precise and accurate analysis, we perform statistical tests on a large number of samples to detect traffic discrimination. This section provides concrete evidence of discrimination in several large ISPs based on routing and traffic content. The next section examines in greater depth the mechanisms and policies used for carrying out traffic differentiation.

4.1 Data processing

Each data point is obtained by sending 200 packets from a probing source host s to a destination IP address d, traversing a target ISP I using packets representing a particular application a. A data point at time interval i is denoted as $l_{\{s,d,I,a,i\}}$ (percentage of lost packets relative to the 200 probing packets) and $d_{\{s,d,I,a,i\}}$ (average delay of 200 delay measurements).

We define two key variables: a path pa which defines the smallest path unit for discrimination analysis and an aggregation element agg which excludes certain variables in the definition of corresponding pa. An agg helps identify the relevance of some factor in discrimination. For example, for application based discrimination analysis, pa=(s, d, I, a) and agg=(s, d, I). To detect whether discrimination exists between applications a_1 and a_2 on the path from s to d in ISP I, we compare the performance of pa_1=(s, d, I, a_1) and pa_2=(s, d, I, a_2). For previous-hop AS based discrimination analysis, pa=(AS_p, P_i, P_e, AS_n) and agg=(P_i, P_e, AS_n). AS_p and AS_n are the previous-hop and next-hop ASes of I respectively. P_i and P_e are the ingress and egress points of I respectively. These notations will be used in the following analysis.

Prior to performing discrimination analysis, we filter measurement noise caused by resource competition on or near a host by identifying high loss rates on many paths that share the same host. We also filter noise caused by ICMP rate limiting by identifying high loss rates that exceed the long-term average plus three times the standard deviation.

4.2 Statistical test to infer discrimination

Assuming random noise has roughly the same impact on the data points measured on any path, we apply statistical tests on several data points to identify consistent performance differences caused by traffic discrimination rather than due to random noise. There are quite a few standard hypothesis tests that compute the statistical significance of the difference between the mean values of two data sets. T-test, the most commonly-used one, requires the data sets under test to follow normal distribution which may not hold for loss rate and delay distributions. So instead, we apply the Wilcoxon signed-rank test [19] and the permutation test [20]. Neither test relies on any assumption of the input data distribution. This is a standard approach for testing the difference between two distributions without any assumptions on the properties of the distributions.

Our input data consists of two sets of data points for the path pair pa_1 and pa_2 respectively, where pa_1 and pa_2 share a common agg. First, we calculate the difference between each pair of data points after each set is sorted numerically: $z_i = x_i - y_i$. For the resulting difference set Z, we test the hypothesis that $mean_z \neq 0$ using the Wilcoxon test. Then we permute half of the data points and apply Wilcoxon test on the permuted set. The permutation tests are repeated 400 times. If both the Wilcoxon and the permutation tests are passed with 95% significance, we determine that discrimination exists between pa_1 and pa_2.

4.3 Characterization of discrimination

We have implemented NVLens on the PlanetLab testbed [21]. We use all the available PlanetLab hosts, roughly 750 of them, as probers covering about 300 distinct sites. It has been fully operational for more than five weeks to monitor 19 ISPs.

Table 2 illustrates the results based on the loss rate metric. Similar results based on the latency metric are omitted due to the lack of space. For application based discrimination, the baseline application for comparison is HTTP. For each path, we also collect the data points for other applications, *e.g.,* BitTorrent, and compare the loss rate with the HTTP loss rate measured during the same period. For previous-hop AS based discrimination, we compare path pairs that share the same agg=(P_i, P_e, AS_n) but from different AS_p.

Table 2 summaries our main findings regarding the absolute number and percentage of path pairs that pass the

[2]The TTLs are not too large to avoid potential complaints from edge networks.

ASN	ISP name	Tier	Application/protocol types				Previous-hop		Next-hop		Same AS
			BT	UDP	Skype	Game	P-P	P-P-AS	P-P	AS-P-P	AS-P-P-AS
209	Qwest	1	10 , 1	0	0	0	8 , 1	36 , 0.2	1 , 0.1	5 , 0.03	6,0.1
701	UUNet		29 , 0.9	**90 , 3.6**	0	0	89 , 3.5	**633 , 3.6**	13 , 0.5	38 , 0.2	92,0.5
1239	Sprint		4 , 0.3	31 , 3.5	3 , 0.2	0	40 , 2.7	315 , 1.1	4 , 0.2	19 , 0.1	0
1668	AOL Transit		0	0	0	1 , 0.5	4 , 1.7	24 , 0.9	0	0	20,0.3
2914	Verio		13 , 1.4	66 , 6.8	18 , 1.5	0	33 , 3.4	110 , 0.4	10 , 1.1	38 , 0.1	0
3356	Level3		0	1 , 0.05	0	0	**109 , 6**	**746 , 1**	2 , 0.1	7 , 0.01	9,0.1
3549	Global Crossing		14 , 1.7	0	0	2 , 0.2	34 , 3.2	293 , 0.6	30 , 3.1	206 , 0.5	0
3561	Savvis		0	1 , 0.05	0	0	16 , 2.7	254 , 1	3 , 0.5	25 , 0.1	33,0.1
7018	AT&T		0	2 , 0.1	0	0	22 , 1	330 , 1	0	0	0
2828	XO	2	0	0	0	0	0	0	0	0	0
2856	British Telecom		0	45 , 4.5	0	0	15 , 1.5	45 , 0.4	2 , 0.2	6 , 0.02	40,1
3257	Tiscali		**221 , 8**	0	17 , 1	0	21 , 3	**184 , 3**	2 , 0.2	6 , 0.1	0
3320	Deutsche Telekom		6 , 0.4	0	0	0	5 , 0.4	26 , 0.2	0	0	11,1
5511	France Telecom		9 , 1	0	29 , 3	0	10 , 1	38 , 0.3	0	0	13,1
6395	Broadwing		0	0	0	0	2 , 0.2	5 , 0.09	0	0	0
6453	Teleglobe		0	68 , 6	0	11 , 1	17 , 1	68 , 0.6	0	0	3,0.2
16631	Cogent		0	0	4 , 0.05	0	70 , 4	213 , 0.8	55 , 3	134 , 0.2	94 , 0.3
6461	AboveNet	3	0	24 , 2.5	0	0	8 , 0.8	37 , 0.4	0	0	0
11537	Abilene		0	0	0	0	0	0	0	0	0

Table 2: Statistical test for loss-based discrimination: discriminated path pairs in absolute number, percentage(%).

statistical test. These two numbers illustrate whether discrimination exists and how widely it is detected in an ISP. Surprisingly, evidence exists for traffic discrimination within backbone ISPs. UUNet, Tiscali, Sprint, Level3, Savvis, and AT&T all have hundreds of path pairs that exhibit previous-hop AS based discrimination. The bold numbers highlight this evidence. Next-hop AS based discrimination is far less prevalent, probably due to the ease of implementation and effectiveness in managing internal resources for the previous-hop based approach. An ingress router can easily mark packets based on their incoming interfaces. There also appears to be application based discrimination, in particular against BitTorrent and UDP traffic. We found one ISP, Tiscali, which exhibits strong evidence of discrimination against BitTorrent traffic. Figure 3 shows significant loss rate difference for the discriminated path pairs: at least 30% of the path pairs have loss rate differences ranging from 3% to 8%.

ISPs usually have incentives to give customers high priority for business reasons. To confirm this claim, for previous-hop based discrimination, we further analyze the relationship between the previous-hop AS and the ISP performing discrimination. We employ the commonly used Gao's relationship inference results [22]. Among the previous-hop discrimination, we found that 51% of path pairs involve ISPs favoring their customers' traffic over peers' traffic. 10% of the path pairs gave traffic from siblings higher priority over customers and peers. We also found many instances of particular peers being given preferential treatment over other peers. For example, among UUNet's peer, Level 3 and Savvis receive better treatment than other peers.

To further confirm that previous-hop discrimination indeed exists, we apply the same statistical tests to path pairs that share the same (AS_p, P_i, P_e, AS_n) using UDP traffic, which should not be affected by previous-hop or next-hop AS based discrimination. The last column in Table 2 provides the absolute number and percentage of such path pairs that pass the tests. In most cases, they are much smaller than the numbers in the previous-hop column, suggesting that the loss rate difference between path pairs are more likely caused by previous-hop AS based discrimination as opposed to random noise.

ASN	% TOS-marked path pairs with discrimination	% discriminated path pairs matching TOS rules
209	2.1	2.9
701	71	45
1239	16	11
1668	80	76
2914	95	89
3356	92	80
3549	81	70
3561	48	35
7018	90	77
2856	56	41
3257	84	59
3320	0	0
5511	60	17
6453	9	11
16631	91	55
6461	9	6

Table 3: Correlation between loss based discrimination and TOS difference.

5 IN-DEPTH ANALYSIS

Some routers mark the Type of Service (TOS) bit in order to provide different levels of service within an ISP. We study to what extent the loss rate discrimination can be explained by the difference in TOS value. Note that our probing packets trigger ICMP time exceeded messages from routers. These messages include the IP header of the original probing packets, which reveals the TOS value of the original probing packets marked by the routers. This allows us to correlate the loss rate difference with TOS difference for any path pair.

While a large TOS value does not always imply high pri-

ority, we assume an ISP has a consistent rule of mapping a TOS value to a fixed priority. Before performing a correlation, we need to determine this rule. Starting with all the path pairs that pass the discrimination tests, we obtain all the distinct TOS values observed in the ISP. We then construct a mapping from TOS value to the priority it stands for. The mapping is constructed in a way to best explain the loss rate difference between all the discriminated path pairs. For example, if TOS value x stands for higher priority, then paths marked with x should experience lower loss rate.

Table 3 illustrates the correlation results between loss rate discrimination and TOS difference. The second column indicates the percentage of TOS-marked path pairs that exhibit the correct loss rate discrimination. And finally, the third column shows the percentage of discriminated path pairs that match the inferred TOS rules. Both percentage numbers are high for a few ISPs, *e.g.,* AS1668, AS2914, AS3356, AS3549, and AS7018, strongly indicating TOS value is used for discriminating against traffic inside these ISPs. We also check the temporal stability of TOS marking and find the marking of 99.9% of the paths does not change within the six-day analysis.

For application based discrimination, we conduct controlled experiments in order to understand how ISPs perform the discrimination. We vary our probing by using a different port, zeroing the application payload, or bypassing the initial control messages. We study the BitTorrent traffic discrimination in Tiscali as an example. We studied the likelihood that the discrimination is performed based on port number. By changing the port from the default BitTorrent port to 80, the number of discriminated path pairs drops by 50%. Zeroing payload or bypassing control messages has a negligible effect.

6 DISCUSSION

Besides detecting neutrality violations, NVLens can further identify the policies used by the ISPs to perform traffic differentiation and reveal other relevant information such as the location of enforcement, time-of-day effect, and relative versus absolute differentiation. The technique can be extended to discover other types of discrimination, *e.g.,* IPSec vs. non-IPSec. Such information can be used by end-systems to make more informed decisions for selecting routes and ISPs, applying encryption or routing through proxies to overcome some of this discrimination.

Even if ISPs are aware of techniques used by NVLens to perform neutrality violation detection, they cannot easily evade our probing. The probe packets are constructed using real traffic traces and are difficult to distinguish from actual data traffic. Unless ISPs perform stateful TCP flow analysis, it is challenging to identify and preferentially treat our probe traffic. In the future, we can further use two-end controlled experiments to mimic the TCP states.

7 CONCLUSION

In this paper we presented the design and implementation of the first deployed system to accurately and scalably detect network neutrality violations performed by backbone ISP networks. Using collaborative probing from end hosts with innovative application-specific probing on carefully selected network paths, we demonstrate the surprising evidence of traffic discrimination carried out by today's backbone ISPs. NVLens has been operational on PlanetLab for five weeks and is capable of monitoring 19 large backbone ISPs simultaneously for neutrality violations detection using loss rate and delay as performance metrics. In addition to detecting network neutrality violation, we perform in-depth analysis to further examine the discrimination policies. Our work demonstrates the feasibility of detecting network neutrality violations in backbone ISPs entirely from end systems and presents an important step to attain more accountability and fairness on today's Internet. To devise countermeasures against ISPs' action of network neutrality violations, detection is an indispensable first step, and our proposed system NVLens is a promising approach.

REFERENCES

[1] J. Crowcroft, "Net neutrality: the technical side of the debate: a white paper," *ACM Computer Communication Review*, 2007.

[2] X. Yang, G. Tsudik, and X. Liu, "A Technical Approach to Net Neutrality," in *Proceedings of ACM HotNets-V, Irvine*, 2006.

[3] I. Avramopoulos, J. Rexford, D. Syrivelis, and S. Lalis, "Counteracting discrimination against network traffic," Tech. Rep. TR-794-07, Princeton University Computer Science, 2007.

[4] I. Avramopoulos and J. Rexford, "Stealth probing: Efficient data-plane security for IP routing," in *Proceedings of USENIX Annual Technical Conference*, 2006.

[5] X. Yang, "Auction, but Don't Block." Work in Progress.

[6] R. Beverly, S. Bauer, and A. Berger., "The Internet's Not a Big Truck: Toward Quantifying Network Neutrality," in *Proceedings of the 8th Passive and Active Measurement (PAM 2007) Conference*, 2007.

[7] G. Lu, Y. Chen, S. Birrer, F. E. Bustamante, C. Y. Cheung, and X. Li, "End-to-end inference of router packet forwarding priority," in *Proc. IEEE INFOCOM*, 2007.

[8] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," *SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 1, pp. 50–60, 2005.

[9] C. Wright, F. Monrose, and G. Masson, "On inferring application protocol behaviors in encrypted network traffic," in *Journal of Machine Learning Research (JMLR): Special issue on Machine Learning for Computer Security*, 2006.

[10] V. S. Kaulgud, "Ip quality of service: Theory and best practices." `www.sanog.org/resources/sanog4-kaulgud-qos-tutorial.pdf`, 2004.

[11] C. S. Inc., "Configuring Priority Queueing." `http://www.cisco.com/en/US/docs/ios/12_0/qos/configuration/guide/qcpq.h%tml`.

[12] "Deep packet inspection." `www.networkworld.com/details/6299.html`.

[13] "Arbor Ellacoya e100 ." `http://www.arbornetworks.com`.

[14] C. Networks, "Complete Packet Inspection on a Chip." `http://www.cpacket.com/`.

[15] R. Mahajan, M. Zhang, L. Poole, and V. Pai, "Uncovering Performance Differences in Backbone ISPs with Netdiff," in *Proceeding of NSDI*, 2008.

[16] S. G. Kolliopoulos and N. E. Young, "Approximation algorithms for covering/packing integer programs," *Journal of Computer and System Sciences*, vol. 71, no. 4, 2005.

[17] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "User-level Internet Path Diagnosis," in *Proceedings of SOSP*, 2003.

[18] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iPlane: An Information Plane for Distributed Services," in *Proc. Operating Systems Design and Implementation*, 2006.

[19] S. Siegel, *Non-parametric statistics for the behavioral sciences*. McGraw-Hill, 1956.

[20] E. S. Edgington, *Randomization tests*. Marcel-Dekker, 1995.

[21] L. Peterson, T. Anderson, D. Culler, and T. Roscoe, "A Blueprint for Introducing Disruptive Technology Into the Internet," in *Proc. of ACM HotNets*, 2002.

[22] L. Gao, "On Inferring Autonomous System Relationships in the Internet," in *Proc. IEEE Global Internet Symposium*, 2000.