Monitoring Network Bias

Gergely Biczók, Whitney Young and Aleksandar Kuzmanovic
Northwestern University
(gbiczok,wbyoung,akuzma)@eecs.northwestern.edu

ABSTRACT

The net neutrality issue is far from being only a theoretical and legal playing field. Internet Service Providers (ISPs) all over the world deploy middleboxes able to filter, shape, and poison certain traffic flows. We propose an end-host based monitoring service capable of detecting such biased network behavior. By using this service users will be able to find out about the discriminatory practices of their ISPs. Building such a system is the necessary first step towards creating an accountable Internet.

Categories and Subject Descriptors

C.2.3 [**Network Operations**]: Network Monitoring

General Terms

Measurement, Experimentation

Keywords

Net neutrality, filtering, shaping, discrimination, Internet Service Provider

1. MOTIVATION

A fundamental question in the net neutrality debate is the extent to which network operators should be allowed to discriminate among Internet packet streams to selectively block or adjust quality of service. Content providers and activists claim that openness of the Internet is needed, while ISPs argue that they have the right to manage traffic on their network and to charge more for value-added services [1]. And while the debate is taking place in public, the battle to enact net neutrality legislation is taking place in Congress, a war is taking place behind the scenes—in the Internet. ISPs all over the world are already installing middle-boxes capable of detecting and actively denying service to VoIP flows, degrading emerging gaming, video and p2p applications [2]. Also, changing/blocking web content for political, social or financial reasons have become common practice.

While the goal of our effort here is *not* to take sides in the net neutrality debate, we argue that independently of different views and future laws, we must have the means to enable network accountability become Internet reality. Since most of us never anticipated that the Internet will ever reach a state in which entire networks intentionally discriminate, degrade or deny service to their own clients, we lack an auditing system that can facilitate such accountability. Thus, a logical first step towards an Internet Audit solution is to design and implement an end-host-based measurement service capable of detecting and exposing discriminatory network practices and elements.

This work is supported by NSF CAREER Award no. 0746360.

Copyright is held by the author/owner(s).
SIGCOMM'08, August 17–22, 2008, Seattle, Washington, USA.
ACM 978-1-60558-175-0/08/08.

2. METHODOLOGY

We identified several discrimination methods that can be used by ISPs to control traffic flowing through their networks, and constructed active probes which are able to detect their use. All of our probes take place between one or more PlanetLab nodes and an end-user's computer residing in an ISP network (Section 3). We differentiate between the directions of given probes as upstream and downstream.

Filtering. All filtering probes share the way of detection: endpoints keep track of sent and received data, and if there is a major difference between the number of packets sent and received, we suspect filters along the path. The key challenge is determining the appropriate filter parameters used by discriminating middleboxes. By sending probes that detect different filtering flavors sequentially we can determine the exact type (or types) of filters used. We aim to cover a wide range of popular networked applications such as p2p filesharing, streaming, gaming, voice-over-IP, etc.

The simplest scenario is *port-based filtering*, when ISPs simply block an incoming or outgoing TCP/UDP port. The probe consists of simply sending multiple packets with random data on the given applications' ports both upstream and downstream. Since applications (particularly p2p file sharing solutions) began to use random port numbers to avoid port-blocking, *signature-based filtering* has emerged. These filters do pattern matching on application-specific byte signatures in the packet payload and mark flows to catch targeted traffic. Fortunately, there are several open-source signature-based filters [3] [4], which we use to compile a list of probes which practically "emulate" real-world applications. *Flow-pattern based* methods were proposed recently. The main idea behind these is detecting correlation between parallel TCP (data transfer) and UDP (overlay management) communication used by p2p applications. We construct probes of parallel flows to multiple endpoints characteristic for p2p systems carrying random data to observe if flow-pattern based filtering is deployed.

Shaping. The basic idea behind designing probes for detecting shapers is to compare download/upload rates or completion times of the targeted application (e.g., BitTorrent, SSL) and a standard application (such as downloading over HTTP/FTP). We argue that consecutive minute-long experiments are satisfactory to determine if there is a significant difference in peak rates and transfer times. For simplicity and feasibility, we "emulate" the targeted application's behavior based on protocol definitions and live traffic traces, instead of installing full-fledged software for every targeted application at every endpoint.

Locating middleboxes. Being able to reveal the location of both filters and shapers significantly adds to our understanding of biased network behavior. We locate discriminative network devices by repeating the same measurements between a client machine and multiple specifically selected PlanetLab nodes. The selection mechanism chooses disjoint testing nodes such that probes travel partially disjoint network paths. By comparing measurement results from these paths, we can detect the presence of middleboxes at the shared path segments—typically in the access network of the studied end-host's ISP.

Miscellaneous. There are additional reported ways for ISPs to put their feet in user traffic flows. Some providers do not allow downloading *.torrent files; this can be easily detected by initiating such a download. Man-in-the-middle TCP RST poisoning can be discovered by observing frequently reseted TCP connections at one or both endpoints. Finally, DNS hijacking of mistyped web addresses can be detected by comparing the requested and returned URL on a client machine.

3. SYSTEM DESIGN

We argue that our system should be easily accessible for most end-users to get a global picture of biased network behavior and upgradeable after deployment to incorporate newly emerging discriminatory techniques. With these requirements in mind we decided to build a *web-based* system. The elements of the proposed system can be seen in Figure 1. Our central server serves the main website to the clients. When a client starts the measurement by clicking on the "Start" button, the central server selects appropriate PlanetLab node(s) that will actually interact with the client through a Java applet. Then active probe traffic is initiated from the client to the PlanetLab node and vice versa. When probes are finished, the client machine reports back the results to the PlanetLab node, which then adds his own report to the client's and send it to the central server for further processing. In the meantime, the PlanetLab node creates a quick report for the client, which is then displayed in the client's web browser.

There are three aspects of the system worth further discussion. First, Java applets loaded over the Internet are prevented from having root access to the client, reading and writing files on the client file system, and from making network connections except to the originating host. We tailored our approach to overcome these restrictions: probes use the standard Java Socket API, results are brief thus can be kept in memory, and connections to multiple PlanetLab nodes are handled in a one-to-one fashion with the central server keeping track of them. Second, as it was mentioned above, to be able to locate discriminative network elements we have to select appropriate PlanetLab nodes intelligently. Currently

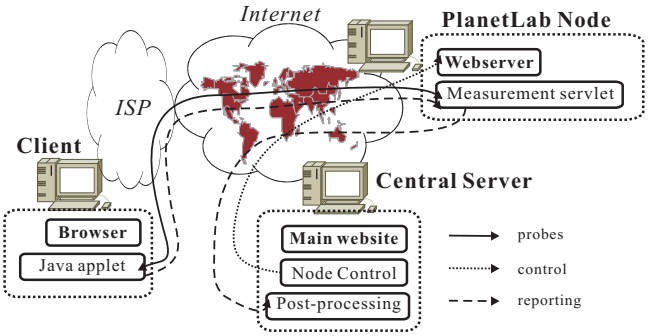

Figure 1: System components

our node selection is based on mapping the client's IP address to a geographic location and assign PlanetLab nodes accordingly. However, we are looking into using iPlane [5] to infer routes between the client and candidate PlanetLab nodes in real time. Third, we use all reports collected and sent to our server to infer network-wide discrimination characteristics. Once we have enough data we can determine typical behavior of different ISPs regarding shaping bandwidth, filtering practices, etc. This will enable us to draw a global map of biased network behavior. In addition, correlating different reports could help us deal with measurement anomalies, erroneous results and client firewalls.

4. RESULTS AND FUTURE WORK

Although we are in the process of implementing the proposed measurement system, we have proof-of-concept results using the proposed probing methods manually at a small scale. As an example, we have encountered BitTorrent shaping several times: while achieving a peak download/upload rate of 30 KBps/5 KBps on well-seeded torrents, we could maintain a steady FTP download rate of 200 KBps at the same time. Shaping was independent of the number of parallel torrents in use and appeared to be on a per user basis. Likewise, we also observed TCP RST poisoning of BitTorrent flows proven by packet traces from both endpoints. We also set up a small testbed, where we validate our probes by using open-source filtering and shaping software. These validation experiments are used to fine-tune our probes before going live with our system. In the near future we plan to publicize our system and conduct a large-scale measurement.

5. REFERENCES

[1] J. M. Peha. The Benefits and Risks of Mandating Network Neutrality, and the Quest for a Balanced Policy. In *Proceedings of TPRC 2006*.

[2] Bad ISPs. `http://www.azureuswiki.com/index.php/Bad_ISPs`.

[3] Application Layer Packet Classifier for Linux. `http://l7-filter.sourceforge.net`.

[4] IPP2P project. `http://ipp2p.org`.

[5] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iPlane: An Information Plane for Distributed Services. In *Proceedings of OSDI 2006*.