Detection of Traffic Discrimination in the Internet

Vinod S. Khandkar and Manjesh K. Hanawal
IEOR, IIT Bombay, Mumbai, India
email:{vinod.khandkar, mhanawal}@iitb.ac.in

***Abstract*—The Internet provides a platform for various commercial activities, and it is essential to ensure that it remains a level playing field for all the players. Several countries have enacted laws such that the Internet remains neutral by prohibiting preferential treatment of traffic of one application or content over the other. However, to enforce such regulations, one needs to detect any violations. In this work, we demonstrate a method to identify non-neutral behavior by comparing the quality of service received by different applications 'when they experience similar transmission as well as network conditions.'**

I. Introduction

Network neutrality is related to 'equal' treatment of the same type of traffic (video, audio) on the Internet irrespective of their origin and destination. Many countries have laws enforcing neutrality, and India is one of them. Telecom Regulatory Authority of India (TRAI) prescribes that any discrimination of the traffic for reasonable network management is allowed. However, any deliberate discrimination of traffic of one application while favoring others of the same type is prohibited. As the Internet evolves as a preferred platform for many commercial activities, the guarantee of high-performance is expected from Internet service providers (ISPs). Often ISPs themselves are content providers (CPs), and would like to ensure higher performance guarantee to their content than that of their competitors, or, in the worst case, can even degrade competitors performance to capture higher market share. For example, ISPs like Airtel and Jio offer their music and TV apps that provide similar content. Our goal is to demonstrate a tool helping users identify any degradation in the quality of service (QoS) for the content due to deliberate differentiation by the ISPs.

Few attempts are made by researchers to detect anomalies related to the net-neutrality violation. Active probing [1]–[8] and passive measurement [9] are two commonly used types of methodologies. Passive measurement refers to the mechanism in which internet traffic is passively monitored and then analyzed for any anomaly. Active probing relates to the method which sends specially crafted traffic packets on the Internet and observes the response of the network. Parameters like port numbers, data contents are varied to form probing packets, and the response to these variations, like inter-arrival times, packet loss, and throughput, are recorded. The anomaly in network behavior is then detected either by seeing patterns within the received data stream or by comparing the performance between two differently crafted streams. The later is called the active differential probing.

While promoting non-discrimination policy towards all packet flowing on the Internet, the net-neutrality regime also allows reasonable traffic management cite. Moreover, regulators all over the world define preferential treatment for many services, such as a medical emergency. Such activities alter the performance of Internet traffic in the same way as congestion or any other network error scenarios and results in momentary traffic differentiation. Failure to distinguish between such a situation from malicious traffic differentiation is a drawback of most of the methods mentioned above. Also, active probes are easily recognized by the network devices as they are very directed, making the network to let them experience favorable quality while manipulating other traffic. The differential probes partially alleviate this issue. However, active probe packet stream creation puts a limitation on their detection capability– the random data in the probe packet limits their applicability where the differentiation is done based on the type of content and its signature.

Another method employed in literature to detect any discrimination is to compare the performance experienced by traffic of service with and without VPN connection [10]. It is available as an App on multiple mobile platforms. The presumption here is that with a VPN connection, an additional layer of encryption encapsulates all related connection parameters making it difficult for the ISPs to classify the traffic and hence cannot apply any discrimination. Then one can compare the performance of traffic with and without VPN connection to identify the differentiation. However, such comparison may not always lead to correct results as VPN flows are likely to get the first-come-first-serve type of treatment as opposed to original streaming content, depending on the load balancing criteria used. It is because the network may treat these VPN flows as different flows than their non-VPN counterparts, and may apply different management rules. Hence it cannot be used as a reference performance for comparing other traffic streams experiencing actual QoS based traffic management.

The variation in the QoS experienced by users for different services could be due to varying levels of congestion experienced by the service on the path they traverse, location of the source, any traffic management, and proprietary dynamic performance adaptation mechanism. It is a natural Internet-borne degradation. Our primary concern is to differentiate between this natural Internet-borne degradation in the QoS and the one deliberately introduced by the ISPs. We propose to leverage the advantages of active probes where probe traffic is the original application traffic transferred from a single dedicated server. Having a common source for all probe traffic forces them to experience the same congestions level, traffic management, when accessed from a client. Further, we ensure that the

978-1-7281-3187-0/20/$31.00 ©2020 IEEE

server applies the same packet transmission policy to all traffic, and the client applies the same traffic management policy to all services, which eliminates the variation in performances due to the dynamic performance adaptation mechanism used by different CPs. Also, we propose a comparison of traffic performances between a pair of applications to determine the application of any malicious traffic management activity.

II. Problem Definition

The task of identifying discrimination on the Internet is delicate. The reason is the Internet-borne degradation of streaming service performance. But most of the time, the varying performances are the direct consequence of dynamic adaptive schemes used by different streaming services. It means that the amount of data sent from the source within the same given time is not always the same for different streaming services under the same conditions or at the source itself.

The data transferred by the multimedia streaming server is a function of many parameters. There can be fixed parameters such as QoS for non-subscribed users and dynamic parameters such as multimedia block length due to change in requested resolution due to available bandwidth conditions or delayed data requests from user-client. It makes the varied amount of data transfer, creating uncertainty in the measurements. These dynamic parameters are mainly related to adaptive streaming protocols and streaming server transmission strategy. So the uncertainty in measurements is mainly due to these two mechanisms. The later problem is reasonably easy to tackle as compared to the first one. The adaptive streaming protocols differ across streaming servers in varied ways. It could be the validity of data on the specific content data server, block sizes due to variation in coding and compression techniques, inter-fetch time of data blocks, server transmission rate due to the selection of the specific type of data block. It is not an exhaustive list but reasonably demonstrates the issue of uncertainty in performance measurement. So the main contributors for this data variations are different streaming data fetch schedules and fetched data size.

The notion of similar network conditions is hypothetical if one considers streaming content from original streaming servers. It is because the Internet is free to route packets from the streaming server through any physical route considering the best-effort delivery policy. The difference in the physical path between source and destination also contributes briefly by making different streams experience different congestion levels. These factors make it impossible to compare the performances of streaming services directly for detecting discrimination. It is thus hard to come to any concrete conclusions by any statistical measurements whether any observed poor QoS is due to deliberate discrimination or natural congestion in the network. We propose to develop techniques to identify whether a poor QoS experienced by users is due to deliberate discrimination and develop a light-weight mobile app that the users can use to ascertain this.

III. Solution approach

Streaming services use proprietary dynamic traffic management policies to give end-users a better experience. These dynamic traffic management could use a different schedule and size for fetching data from the servers. Hence a direct comparison of QoS experienced by an end-user for different services is not useful to distinguish deliberate discriminations from legitimate QoS adaptations. The solution to the above problem is unifying traffic management policy for all the services and host them on a shared test server for doing a replay. We host traffic of all the services of interest on this server by collecting it from the original servers. For example, the server stores a chunk of data associated with a YouTube video as YouTube traffic. The server sends traffic associated with services requested by a client using the same traffic management policy. Our client-server architecture is shown in Figure 1.

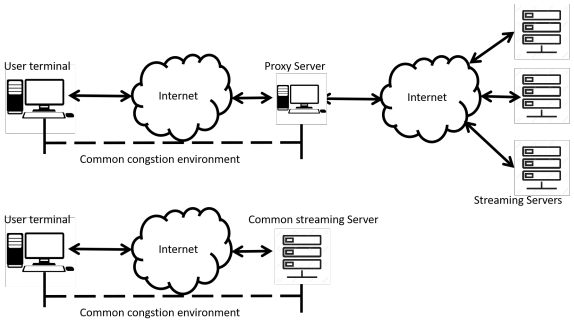

Fig. 1. Unifying the Internet environment using common streaming server

A. A common traffic management policy

Streaming service uses dynamic adaptive algorithms at both client and server. The policy at the client governs the data fetch schedule. Many times they are partially controlled by the steaming server by explicitly providing a data download schedule like Dynamic Adaptive Streaming for HTTPS (DASH). Unification of the fetch schedule requires making DASH identical for all types of application traffic under test. Let us look at this data fetch schedule. The streaming of media content involves media client on host machine sending series of HTTP GET commands for downloading required content. The data fetch schedule consists of the URLs requested in HTTP GET command along with the inter-command delays. The first step is to make clients of all streaming services send HTTP GET requests to the respective server with the same schedule. Our experimental results show that all streaming servers respond to such HTTP GET requests with either proper redirection information or the content itself. So, the user client sends HTTP GET commands immediately after the download corresponding to the previous GET command is done. This mechanism effectively removes inter-command duration. It is part of the adaptive streaming strategy and creates a performance comparison issues. We remove any timing relationships while fetching data using GET commands to unify client-side DASH of all streaming services.

B. A common packet transmission policy

The streaming server responds with streaming data corresponding to the requested data segment. From a performance variation perspective, the operation of the streaming server varies in terms of requested data segment sizes and data burst sent on the Internet. The different number of HTTP GET requests from user clients for the same amount of data is the direct consequence of different data segment size. Each HTTP GET request is associated with a finite amount of delay due to processing. It means delays due to different segment sizes differ for streaming services. The next step towards unifying the server-side environment is to have the same number of data segments for the given amount of streaming data. Thus the performances of streaming services get affected by the same amount of processing delay. Next, the burst of data sent from the server results in high throughput activity in the channel. Thus attracts the attention of network traffic management devices. The necessary traffic management is the outcome of it. Different burst sizes lead to the application of different traffic management applications and thus introduces performance variations. The same burst size for all streaming services makes streams to use similar Internet bandwidth. The use of the same adaptive transmission policy results in the same burst sizes for all services.

The next factor affecting end-to-end streaming performance is TCP implementation. There are different TCP flavors developed over time, having different characteristics. Each streaming server employs one of these TCP implementations. This selection helps to adapt best to the Internet condition as per streaming strategy. But, it makes the transmitter congestion window to differ across servers. It results in variation in experienced throughput at the user client-side. This aspect is entirely under the control of the streaming service provider. The unification of TCP implementation sees the same kind of re-transmissions and a similar amount of congestion window progression for all streaming services.

Our uniform traffic management policy makes the traffic generated by different services experience the same environment as they traverse the Internet. Thus their QoS becomes comparable. The traffic differentiation experienced by similar services is detected using the throughput comparison based algorithm.

IV. Technology Demonstration

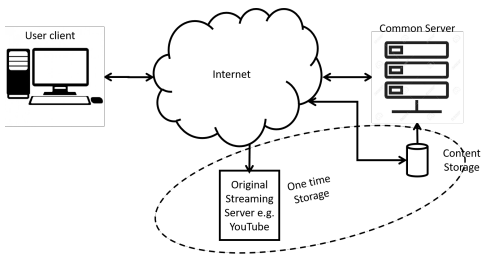

Fig. 2. End-to-end performance measurement setup

The system, as shown in Fig. 2 consists of a test server and a user client. The user client is in the form of an Android App but not yet available on Google or Apple mobile platforms. It needs no special permissions. It uses an Internet connection on the host device to communicate with the test server. The demonstration includes the procedure to perform the traffic differentiation detection test. The tests use different data rates on the server-side. Tests with different data rates at the server are to demonstrate two facts. First, to show that differentiation detection algorithm works properly at higher as well as lower data rates. Second, it shows the importance of the client bandwidth management algorithm at the server-side. The user starts the test by selecting the streaming service under the test. The user also needs to provide one or more similar services for comparison from the list of given similar services. During the test, the user client downloads 20Mb data per service from the test server. It does not consume this data in the form of user playback. It extracts various traffic parameters, such as burst size, burst timings, of traffic streams. It uses these extracted parameters for calculation of the application level cumulative throughput. It then compares the throughput for different services using a novel traffic differentiation detection algorithm. The system provides the results of the comparison to the user at the end of the test. It is in the form of the differentiation verdict for the service under consideration. Currently, the system supports only stored streaming services like movie streaming services. The addition of support for more service types and services is future work.

References

[1] M. Dischinger, A. Mislove, A. Haeberlen, and K. P. Gummadi, "Detecting bittorrent blocking," in *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, Oct. 2008, pp. 3–8.

[2] P. Kanuparthy and C. Dovrolis, "Shaperprobe: End-to-end detection of isp traffic shaping using active methods," in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, pp. 473–482.

[3] C. B. R. Ravaioli, G. Urvoy-Keller, "Towards a general solution for detecting traffic differentiation at the internet access," in *2015 27th International Teletraffic Congress*, Sep. 2015, pp. 1–9.

[4] 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 *Proceedings International Conference on Computer Communications (INFOCOM)*, 2007, pp. 1784–1792.

[5] (Jun.) Ookla, measuring and understanding broadband : Speed, quality and application. [Online]. Available: https://www.ookla.com/docs/UnderstandingBroadbandMeasurement.pdf

[6] M. Dischinger, M. Marcon, S. Guha, K. Gummadi, R. Mahajan, and S. Saroiu, "Glasnost: Enabling end users to detect traffic differentiation," in *7th USENIX Conf. Networked Systems Design and Implementation*, Apr. 2010, pp. 27–27.

[7] U. Weinsberg, A. Soule, and L. Massoulie, "Inferring traffic shaping and policy parameters using end host measurements," in *Proceedings of IEEE INFOCOM*, 2011.

[8] P. K. C. Dovrolis, "Diffprobe: Detecting isp service discrimination," in *Proceedings of IEEE INFOCOM*, 2010.

[9] M. T. M. M. N. F. M. Ammar, "Detecting network neutrality violations with causal inference," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies (CoNEXT)*, 2009.

[10] A. Molavi Kakhki, A. Razaghpanah, A. Li, H. Koo, R. Golani, D. Choffnes, P. Gill, and A. Mislove, "Identifying traffic differentiation in mobile networks," in *Proceedings of the 2015 Internet Measurement Conference*, Oct. 2015, pp. 239–251.