Traffic Differentiation on Cellular Data Networks

Arash Molavi Kakhki, Abbas Razaghpanah, Phillipa Gill, Alan Mislove, Dave Choffnes

Abstract

As wireless providers offer faster and more widely available cellular data to their customers, more mobile applications emerge that utilize this to add more services to the mobile platform or just simply improve on the quality of existing ones. It is not clear how traffic from said services is treated while traversing boundaries of these networks.

The goal of this research is to detect traffic differentiation in cellular data networks. Traffic differentiation is defined as tampering with the performance of the network done by the ISP in any shape or form.

1. INTRODUCTION

Service differentiation might be done for a variety of reasons. As an example, a wireless provider might limit the performance of third-party VoIP or video calling services (or any other competing product or service) by introducing delays or reducing transfer rates to discourage users from using them instead of the ones provided by the wireless provider. Also, a wireless provider might limit or disable traffic flow for file sharing applications, video streaming services, or other kinds of relatively ”network-intensive” applications.

These policies are often not disclosed with users or buried deep within fair usage policy sections of service contracts. T-Mobile UK state in their data plan’s fine print that VoIP is not allowed on their monthly cellular data plan [3], Straight Talk does not allow streaming, downloading, or uploading of uninterrupted videos as well as peer-to-peer (P2P) file sharing [2]. Our goal is to perform a number of tests on cellular networks to determine if they differentiate services.

Service differentiation can be done based on host, port, and/or packet payload. As an example, an ISP could block all the connections that use port `4070` to block Spotify, reduce bandwidth for all traffic to and from `*.dropbox.com` for Dropbox, and block all the packets that look like BitTorrent traffic (regardless of host/port). Due to the approach of this research, the latter two can be detected using this method.

Previous work on detecting service differentiation worked on home ISPs and not mobile networks, this is covered in Section 2. The Glasnost project worked on detecting differentiation of BitTorrent traffic in home ISPs using a browser applet. Our approach will test cellular networks for differentiation on a set of suspected services using Meddle VPN. Our methodology is explained in section 3.

2. PREVIOUS WORK

Glasnost and Meddle. In a previous effort to detect service differentiation, the Glasnost project made a web-based tool that users could use to see if their ISPs differentiates their traffic. Glasnost detects service differentiation for BitTorrent and does so by comparing metrics for a ”reference” flow and ”BitTorrent” flow over BitTorrent ports.

Glasnost makes a connection to a measurement server’s BitTorrent port and sends BitTorrent traffic, then it sends the same amount of data to the same server, only this time the data does not look like BitTorrent data. This process is repeated many times both on on BitTorrent ports and on other ports. Then the metrics calculated are compared to see if BitTorrent flow is being treated differently. This way, Glasnost can identify differentiation based on port and/or content.

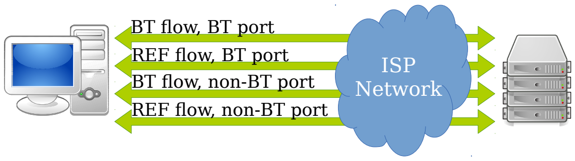


Figure 1: Schematic view of how glasnost works.

This measurement can be performed by logging into Glasnost test page and loading the Java^{TM}applet. This design will not allow mobile devices to run the test as mobile devices can not run Java applets in their web browsers. Additionally, Glasnost is designed only to detect BitTorrent traffic differentiation.

3. METHODOLOGY

The method used here to detect differentiation is quite similar to that of Glasnost. To see if an ISP treats a certain type of traffic differently, we first record all packets transmitted/received during a sample run of that service, replay it on the cell network using a controlled server and then repeat this process (replaying the recorded trace) on an encrypted channel. This process is repeated 5 times for each channel (encrypted and unencrypted). Finally, if the ISP treats traffic differently, metrics calculated for each channel should be significantly different (i.e. significantly more delay on Skype traffic when replaying on an unencrypted channel). Figure 2 is a rough sketch of what happens during a test.

As stated before, this approach resembles that of Glasnost, meaning that it also can only detect differentiation based on port/packet payload. The reason for this is that it is not possible to send/receive recorded data to/from the original servers (e.g. Skype servers). So if the ISP differentiates based on host, it will go undetected. But ISPs usually differentiate based on port/packet payload [1].

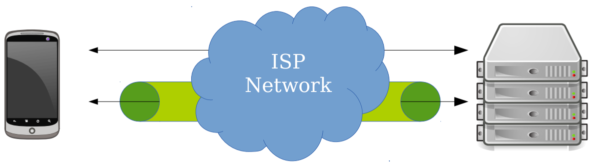


Figure 2: Schematic view of how our project works.

3.1 Services

A number of services were selected to test for differentiation on wireless networks. These tests each represented a category that was suspected to be differentiated by wireless providers.

For the VoIP category, Skype was chosen because it is a widely used service. For file sharing and P2P, Dropbox (upload and download) and BitTorrent were chosen respectively. Netflix, Hulu+ and YouTube (upload and download) represent video, and Spotify and Pandora represent audio in the streaming category.

3.2 Recording Traces

Packet traces for sample sessions for each service had to be recorded first in order to be replayed later over a cellular network. In order to do so, a mobile phone was connected to a VPN (in WiFi mode) server that recorded all of the packets in pcap format using `tcpdump`. When connected to the VPN, an action that required network interaction on the phone would be initiated and after enough data was transferred, the action would be interrupted and the VPN connection would be disconnected.

A few qualities had to be verified about reference service trace: the reference WiFi network on which the packets were recorded should be well provisioned itself (i.e. no service differentiation mechanism should be in place), so the traces were recorded on campus WiFi network. Additionally, tested applications shouldn't behave differently on WiFi compared to when they are on cellular network. This was confirmed for all tested applications by recording traces once on a WiFi network and an HSDPA+ network with roughly the same bandwidth.

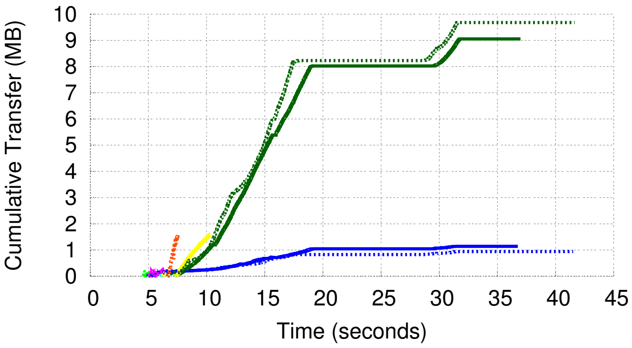


Figure 3: Sequence number plots for connections made during Netflix two sessions on WiFi (solid) and HSDPA+ (dashed) look similar.

3.3 Replay

To replay the trace, a copy of the pcap file is sent to the server and a copy is kept on the client. The client will establish each connection to the server on the same port. The server will do the same and only sends packets once it has seen the packets it expects to have seen based on the pcap file it has. A pcap of this replay will be kept on the client for analysis.

For the replay we needed to decide if we wanted to preserve the inter-packet timing that would reflect application level flow control. Because the reference packet traces were recorded on a WiFi network with significantly higher bandwidth, this would have no effect on the outcome of the tests because this wouldn't affect the differentiation.

3.4 Noise Packets

A smartphone that is connected to the Internet will always send and receive packets due to the services and applications that are running in the background (e.g. mail clients, update agents) so it was decided not to eliminate those packets because they were a natural part of what happens during a session.

Although for the plots, the connections shown are the ones with the largest amount of transferred bytes.

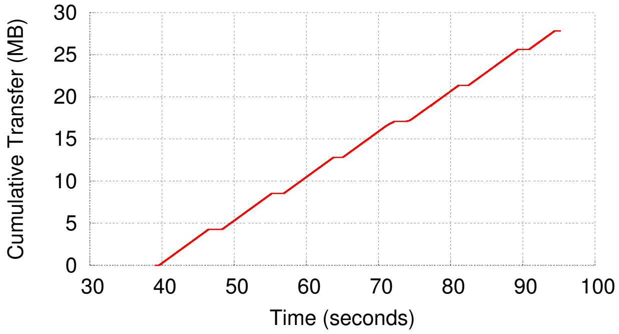


Figure 4: Sequence number plot for a connection made during a Dropbox upload session on WiFi showing application-level flow control.

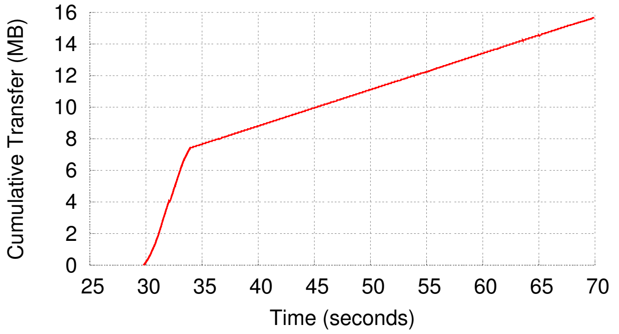


Figure 5: Sequence number plot for a connection made during a YouTube video session on WiFi showing application-level flow control.

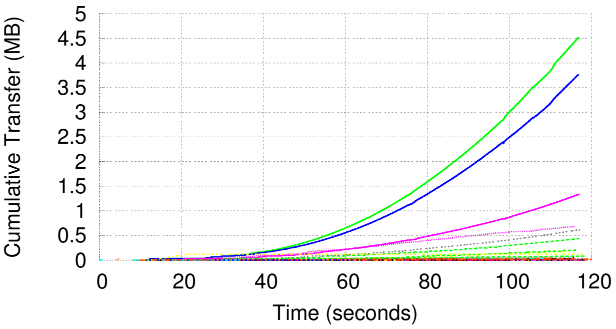
If the connection's total bytes transferred during a session multiplied by a threshold is larger than or equal to the connection with largest number of bytes transferred in that session, it will be in the plot. Figure 6 demonstrates this method.

3.5 Metrics

Metrics that are calculated include round-trip times (RTT), throughput, and loss rate. These metrics are calculated for both channels (encrypted and unencrypted) and compared to detect differences.

In cases that there are statistically significant differences between the metrics for a service, a differentiation mechanism is suspected to exist. This can be further investigated by other methods (e.g. repeating the test). Figure 7 shows an example of this.

Note that some difference in throughput between encrypted and unencrypted replays is expected (and observed, Figure 8) because at the same bandwidth, a VPN connection will add overhead to the packets. Another reason for this maybe the processing carried out by the replay scripts to verify packets using hash functions.



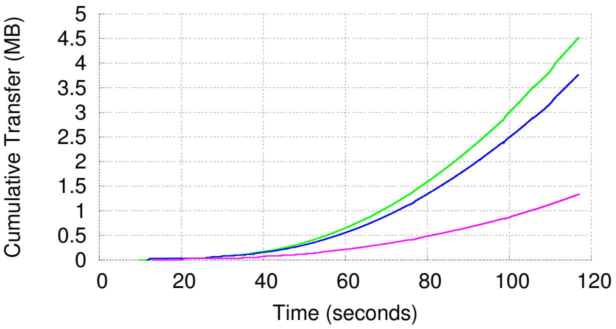


Figure 6: Sequence number plot for connections made during a BitTorrent session on WiFi. Top: without a cut-off threshold, Bottom: With a cut-off threshold of 4

4. RESULTS

Tests were performed on a number of well-known wireless providers in north America and results indicated that differentiation mechanisms are not in fact present in those networks.

Also, as mentioned before, significantly different metrics in a network, while rising suspicion about its treatment of a certain type of traffic, will be inconclusive until complemented by further testing.

Table 1 shows test results from Verizon. These results show that in fact there isn't a differentiation mechanism in their network for any of the tested services. Loss rates for all of the tests were within error margins.

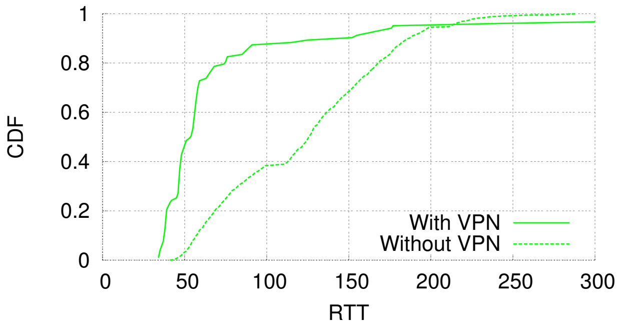


Figure 7: CDF for RTT samples of YouTube upload sessions on AT&T network show higher RTTs when not encrypted.

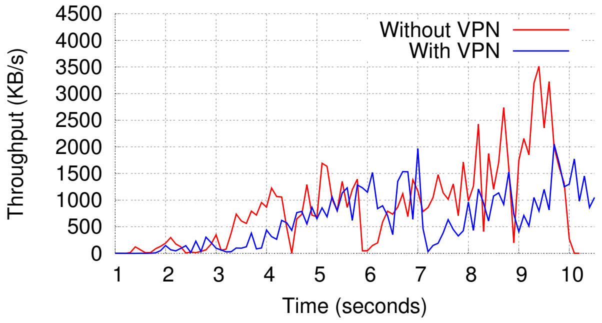


Figure 8: Throughput calculated over 0.1 second intervals for a Spotify session on AT&T. Throughput for the VPN connection is generally lower than that of the unencrypted connection.

5. FUTURE WORK

Currently, the testing and measurement is done on a laptop tethered to a phone by a researcher. The next step would be to implement a cell phone application that does this experiment automatically so that people can freely download the app and perform measurements themselves. This way, more data will become available and the results would be more dependable for the existing cellular networks, as well as giving data to detect differentiation on newer ones.

Additionally, such platform will reveal possible differentiation variations on the same network in different regions.

App	Throughput (KB/s)		Loss (%)	
	No VPN (avg, stdev)	VPN (avg, stdev)	No VPN (avg, stdev)	VPN (avg, stdev)
YT(DL)	(103.74, 31.16)	(99.85, 35)	(0.81, 0.06)	(0.86, 0.13)
YT(UL)	(114.52, 6.05)	(117.37, 8.78)	(0.03, 0.01)	(0.05, 0.01)
DB(DL)	(155.09, 32.42)	(148.1, 44.95)	(0.79, 0.31)	(0.88, 0.38)
DB(UL)	(115.07, 7.31)	(120.25, 5.85)	(0.07, 0.02)	(0.09, 0.01)
SPTFY	(123.91, 40.19)	(127.16, 45.96)	(0.83, 0.08)	(0.73, 0.07)
NFLX	(122.81, 28.42)	(132.26, 33.55)	(0.97, 0.03)	(0.99, 0.15)

Table 1: Average and standard deviation for 4 apps *(YT: YouTube, DB: Dropbox, SPTFY: Spotify, NFLX: Netflix; UL=Upload, DL=Download) on Verizon. The differences in performance are within the noise, indicating no service differentiation.*

6. REFERENCES

[1] M. Dischinger, M. Marcon, S. Guha, K. P. Gummadi, R. Mahajan, , and S. Saroiu. Glasnost: Enabling end users to detect traffic differentiation. In *NSDI*, 2010.

[2] Straight Talk Terms and Conditions, 2013. `http://www.straighttalk.com/`.

[3] T-Mobile UK Monthly Plans Legal Information, 2013. `http://www.t-mobile.co.uk/shop/terms-and-conditions/pay-monthly/`.