Identifying Traffic Differentiation in Mobile Networks

Arash Molavi Kakhki§, Abbas Razaghpanah†, Anke Li†, Hyungjoon Koo†, Rajesh Golani†, David Choffnes§, Phillipa Gill†, Alan Mislove§

Northeastern University§, Stony Brook University†

This work is generously supported in part by a Google Faculty Research Award
Introduction
Introduction

Traffic differentiation
Traffic differentiation
selectively changing the performance of network traffic
Introduction

Traffic differentiation
selectively changing the performance of network traffic

Reasons for differentiation:
- traffic engineering
- bandwidth management
- business reasons
- unfair business practices affecting completing technologies
Introduction

Traffic differentiation
selectively changing the performance of network traffic

Reasons for differentiation:
• traffic engineering
• bandwidth management
• business reasons
• unfair business practices affecting completing technologies

Do certain types of network traffic receive better (or worse) performance?
Related work

Closely related work:

• Glasnost
• NetPolice
• NANO
• Bonafide
• …

Limitations:

• Limited protocols
• Limited application
• Limited environments
• Synthetic traffic (not scalable)
• Unreliable performance baseline
• …
Goals

Reliably detect differentiation in cellular networks

- On any application traffic without access to source
- Without requiring root privileges or OS modifications
- With few assumptions about traffic characteristics or packet shaper implementations

Our approach is the only known way to test differentiation from non-rooted mobile devices
How is differentiation done?
How is differentiation done?

Differentiation in practice consists of
How is differentiation done?

Differentiation in practice consists of

1. Matching traffic on a **classifier** for a class
How is differentiation done?

Differentiation in practice consists of

1. Matching traffic on a **classifier** for a class
2. Implementing some **policy** on that class (shape, policing, block, modify, …)
How is differentiation done?

Differentiation in practice consists of

1. Matching traffic on a classifier for a class
2. Implementing some policy on that class (shape, policing, block, modify, …)

Identifying differentiation requires
How is differentiation done?

Differentiation in practice consists of

1. Matching traffic on a **classifier** for a class
2. Implementing some **policy** on that class (shape, policing, block, modify, …)

Identifying differentiation requires

1. Some way to **trigger** a classifier
How is differentiation done?

Differentiation in practice consists of

1. Matching traffic on a **classifier** for a class
2. Implementing some **policy** on that class (shape, policing, block, modify, …)

Identifying differentiation requires

1. Some way to **trigger** a classifier
2. Statistical techniques to **reveal** policies
Assumptions

What triggers classification?
Assumptions

What triggers classification?

*We don’t know.*
Assumptions

What triggers classification?

We don’t know.

We assume that they might trigger on:

- **IP addresses**
- **ports**
- **payload signatures**
- total **number of connections**
- total **bandwidth**
- **time of day**

This is consistent with online manuals for boxes with deep packet inspection and shaping capabilities.
Roadmap

1. Methodology: how do we do it?
2. Validation: does it actually work?
3. Deployment and findings
Roadmap

1. Methodology: how do we do it?

2. Validation: does it actually work?

3. Deployment and findings
Method: Record & Replay
(at 10,000 feet)
Method: Record & Replay
(at 10,000 feet)
Method: Record & Replay
(at 10,000 feet)
Method: Record & Replay
(at 10,000 feet)
Method: Record & Replay
(at 10,000 feet)
Method: Record & Replay
(at 10,000 feet)
Method: Record & Replay
(at 10,000 feet)

Record:

- **end user**
- **VPN TUNNEL**
- **VPN proxy**
- **NETFLIX**

arash@ccs.neu.edu
Method: Record & Replay
(at 10,000 feet)

Record:

end user

VPN TUNNEL

VPN proxy

end user

NETFLIX

NETFLIX
Method: Record & Replay
(at 10,000 feet)

Record:

Replay:
Method: Record & Replay
(at 10,000 feet)

Record:

End user

VPN TUNNEL

VPN proxy

Replay:

End user

Replay server
Method: Record & Replay
(at 10,000 feet)

Record:

Replay server

VPN TUNNEL

VPN proxy

end user

end user

VPN TUNNEL

arash@ccs.neu.edu
Method: Record & Replay
(at 10,000 feet)

Record:

VPN TUNNEL
end user

Replay:

VPN proxy
end user

Replay server

arash@ccs.neu.edu
Method: Record & Replay
(at 10,000 feet)

Record:

VPN TUNNEL

Replay:

VPN proxy

Shaper

end user

end user

Replay server
Method: Record & Replay
(at 10,000 feet)

Record:

VPN TUNNEL

Replay server

Shaper

VPN proxy

Replay:

end user

end user

NETFLIX

arash@ccs.neu.edu
Method: Record & Replay (at 10,000 feet)

Record:

VPN TUNNEL

end user

Replay:

VPN TUNNEL

Shaper

end user

Replay server

arash@ccs.neu.edu
Method: Record & Replay
(at 10,000 feet)

Record:

VPN TUNNEL

VPN proxy

end user

end user

Replay:

VPN TUNNEL

Replay server
Method: Record & Replay
(at 10,000 feet)

Record:

VPN TUNNEL

end user

VPN proxy

Replay:

VPN TUNNEL

end user

Replay server
Method: Record & Replay
(at 10,000 feet)

Record:
- End user
- VPN Tunnel
- VPN proxy

Replay:
- End user
- VPN Tunnel
- Replay server
- Analyzer
Method: Record & Replay
(at 10,000 feet)

Record:

Replay:

VPN TUNNEL

VPN proxy

end user

VPN TUNNEL

end user

Replay server

Analyzer
Roadmap

1. Methodology: how do we do it?
   - Record and Replay

2. Validation: does it actually work?

3. Deployment and findings
Roadmap

1. Methodology: how do we do it?
   - Record and Replay

2. Validation: does it actually work?

3. Deployment and findings
Validation
1. Do replays look like original traffic?
   • Traffic’s characteristics
   • From shapers’ perspective
Validation

1. Do replays look like original traffic?
   - Traffic’s characteristics
   - From shapers’ perspective

2. Can the analyzer detect differentiation?
Validation

1. Do replays look like original traffic?
   - Traffic’s characteristics
     - From shapers’ perspective

2. Can the analyzer detect differentiation?
Validation

1. Do replays look like original traffic?
   - Traffic’s characteristics
   - From shapers’ perspective

2. Can the analyzer detect differentiation?

![Graph showing cumulative transfer vs. time for original and replay YouTube traffic recorded on Verizon]
Validation

1. Do replays look like original traffic?
   - Traffic’s characteristics
   - From shapers’ perspective

2. Can the analyzer detect differentiation?

Paper explains how replays preserve:
- Logical and time dependencies
- Happens-before relationship
- Protocols, ports, payload, …
Validation

1. Do replays look like original traffic?
   - Traffic’s characteristics
   - From shapers’ perspective

2. Can the analyzer detect differentiation?
Validation

1. Do replays look like original traffic?
   - Traffic’s characteristics
   - From shapers’ perspective

2. Can the analyzer detect differentiation?
Validation

1. Do replays look like original traffic?
   - Traffic’s characteristics
   - From shapers’ perspective

2. Can the analyzer detect differentiation?

Netflix:
Validation

1. Do replays look like original traffic?
   - Traffic’s characteristics
   - From shapers’ perspective

2. Can the analyzer detect differentiation?

Netflix:
Validation

1. Do replays look like original traffic?
   - Traffic’s characteristics
   - From shapers’ perspective

2. Can the analyzer detect differentiation?

Netflix:
Validation

1. Do replays look like original traffic?
   • Traffic’s characteristics
   • From shapers’ perspective

2. Can the analyzer detect differentiation?

Skype:
Validation

1. Do replays look like original traffic?
   - Traffic’s characteristics
   - From shapers’ perspective

2. Can the analyzer detect differentiation?

Skype:
Validation

1. Do replays look like original traffic?
   - Traffic’s characteristics
   - From shapers’ perspective

2. Can the analyzer detect differentiation?
Validation

1. Do replays look like original traffic?
   - Traffic’s characteristics
   - From shapers’ perspective

2. Can the analyzer detect differentiation?
Detecting Differentiation
(not straight forward)
Detecting Differentiation
(not straight forward)
Detecting Differentiation
(not straight forward)
Detecting Differentiation
(not straight forward)
Detecting Differentiation
(not straight forward)
Detecting Differentiation
(not straight forward)

Rate vs. Time

Region 1 (below avg)

Always detect differentiation
Detecting Differentiation
(not straight forward)

Region 1
(below avg)

Always detect differentiation
Detecting Differentiation
(not straight forward)

Rate
Time
Average
Maximum
Shaping rate
Region 1 (below avg)

Always detect differentiation
Detecting Differentiation
(not straight forward)

Region 1 (below avg)

Region 3 (above max)

Rate

Time

Average

Maximum

Never detect differentiation

Always detect differentiation

arash@ccs.neu.edu
Detecting Differentiation
(not straight forward)

Region 3
(above max)

Region 1
(below avg)

Rate

Time

Average

Maximum

Never detect differentiation

Alway detect differentiation

arash@ccs.neu.edu
Detecting Differentiation
(not straight forward)

Rate vs Time

Region 1 (below avg)
- Always detect differentiation

Region 3 (above max)
- Never detect differentiation

Shaping rate

Average

Maximum

arash@ccs.neu.edu

IMC 2015
Detecting Differentiation
(not straight forward)

- Region 1 (below avg): Alway detect differentiation
- Region 2 (between avg and max): Consistently detect or not detect differentiation
- Region 3 (above max): Never detect differentiation
Detecting Differentiation
(not straight forward)
Detecting Differentiation
(not straight forward)

1. Glasnost
Detecting Differentiation

(not straight forward)

1. Glasnost
   • Looks at max throughput
Detecting Differentiation

(not straight forward)

1. Glasnost
   - Looks at max throughput
   - Detects differentiation in R3
Detecting Differentiation
(not straight forward)

1. Glasnost
   • Looks at max throughput
   • Detects differentiation in R3

2. NetPolice
Detecting Differentiation
(not straight forward)

1. Glasnost
   • Looks at max throughput
   • Detects differentiation in R3

2. NetPolice
   • Uses Two-sample KS test
Detecting Differentiation
(not straight forward)

1. Glasnost
   • Looks at max throughput
   • Detects differentiation in R3

2. NetPolice
   • Uses Two-sample KS test
Detecting Differentiation
(not straight forward)

1. Glasnost
   • Looks at max throughput
   • Detects differentiation in R3

2. NetPolice
   • Uses Two-sample KS test
Detecting Differentiation  
(not straight forward)

1. Glasnost  
   • Looks at max throughput  
   • Detects differentiation in R3

2. NetPolice  
   • Uses Two-sample KS test
Detecting Differentiation
(not straight forward)

1. Glasnost
   • Looks at max throughput
   • Detects differentiation in R3

2. NetPolice
   • Uses Two-sample KS test
   • Sensitive to small changes not due to differentiation
Detecting Differentiation
(why KS2 fails)
Detecting Differentiation
(KS2 vs. Area test)
## Detecting Differentiation

![Diagram with components: Replay client, Router, Packet shaper, Gateway, Replay server]

<table>
<thead>
<tr>
<th>App</th>
<th>NetPolice</th>
<th>Area test</th>
<th>Glasnost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R1</td>
<td>R2</td>
<td>R3</td>
</tr>
<tr>
<td>Netflix</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YouTube</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hangout</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skype</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Percentage detected as differentiation
## Detecting Differentiation

<table>
<thead>
<tr>
<th>App</th>
<th>NetPolice</th>
<th>Area test</th>
<th>Glasnost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R1</td>
<td>R2</td>
<td>R3</td>
</tr>
<tr>
<td>Netflix</td>
<td>100</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>YouTube</td>
<td>100</td>
<td>67</td>
<td>0</td>
</tr>
<tr>
<td>Hangout</td>
<td>100</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>Skype</td>
<td>100</td>
<td>55</td>
<td>0</td>
</tr>
</tbody>
</table>

Percentage detected as differentiation
## Detecting Differentiation

Percentage detected as differentiation

<table>
<thead>
<tr>
<th>App</th>
<th>NetPolice</th>
<th>Area test</th>
<th>Glasnost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R1</td>
<td>R2</td>
<td>R3</td>
</tr>
<tr>
<td>Netflix</td>
<td>100</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>YouTube</td>
<td>100</td>
<td>67</td>
<td>0</td>
</tr>
<tr>
<td>Hangout</td>
<td>100</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>Skype</td>
<td>100</td>
<td>55</td>
<td>0</td>
</tr>
</tbody>
</table>

**Diagram:**
- **Replay client**
- **Router**
- **Packet shaper**
- **Gateway**
- **Replay server**
Detecting Differentiation

<table>
<thead>
<tr>
<th>App</th>
<th>NetPolice</th>
<th></th>
<th>Area test</th>
<th></th>
<th>Glasnost</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R1</td>
<td>R2</td>
<td>R3</td>
<td>R1</td>
<td>R2</td>
<td>R3</td>
</tr>
<tr>
<td>Netflix</td>
<td>100</td>
<td>65</td>
<td>0</td>
<td>100</td>
<td>28</td>
<td>0</td>
</tr>
<tr>
<td>YouTube</td>
<td>100</td>
<td>67</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hangout</td>
<td>100</td>
<td>40</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Skype</td>
<td>100</td>
<td>55</td>
<td>0</td>
<td>100</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

Percentage detected as differentiation

arash@ccs.neu.edu
# Detecting Differentiation

<table>
<thead>
<tr>
<th>App</th>
<th>NetPolice</th>
<th>Area test</th>
<th>Glasnost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R1</td>
<td>R2</td>
<td>R3</td>
</tr>
<tr>
<td>Netflix</td>
<td>100</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>YouTube</td>
<td>100</td>
<td>67</td>
<td>0</td>
</tr>
<tr>
<td>Hangout</td>
<td>100</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>Skype</td>
<td>100</td>
<td>55</td>
<td>0</td>
</tr>
</tbody>
</table>

Percentage detected as differentiation
Roadmap

1. Methodology: how do we do it?
   • Record and Replay

2. Validation: does it actually work?
   • Replays maintain features of application traffic
   • Shapers classify replays correctly
   • Area test performs well in all 3 shaping regions

3. Deployment and findings
Assumptions

What triggers classification?

*We don’t know.*

We assume that they might trigger on:

- **IP addresses**
- **ports**
- **payload signatures**
- total **number of connections**
- total **bandwidth**
- **time of day**

This is consistent with online manuals for DPI boxes.
Assumptions

What triggers classification?

*We don’t know.* Let’s find out!

We assume that they might trigger on:

- IP addresses
- ports
- payload signatures
- total number of connections
- total bandwidth
- time of day

This is consistent with online manuals for DPI boxes.
What triggers classification?

[Diagram showing a network with YouTube, Router, Packet shaper, Gateway, and Replay server.]
What triggers classification?

<table>
<thead>
<tr>
<th>Original IPs</th>
<th>Original Ports</th>
<th>Other changes</th>
<th>Services detected as</th>
</tr>
</thead>
<tbody>
<tr>
<td>✅</td>
<td>✅</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>✗</td>
<td>✅</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>✅</td>
<td>✗</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>✗</td>
<td>✗</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
What triggers classification?

<table>
<thead>
<tr>
<th>Original IPs</th>
<th>Original Ports</th>
<th>Other changes</th>
<th>Services detected as</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>✔</td>
<td>-</td>
<td>YouTube</td>
</tr>
<tr>
<td>✗</td>
<td>✔</td>
<td>-</td>
<td>YouTube</td>
</tr>
<tr>
<td>✔</td>
<td>✗</td>
<td>-</td>
<td>YouTube</td>
</tr>
<tr>
<td>✗</td>
<td>✗</td>
<td>-</td>
<td>YouTube</td>
</tr>
</tbody>
</table>
What triggers classification?

<table>
<thead>
<tr>
<th>Original IPs</th>
<th>Original Ports</th>
<th>Other changes</th>
<th>Services detected as</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>✔</td>
<td>-</td>
<td>YouTube</td>
</tr>
<tr>
<td>❌</td>
<td>✔</td>
<td>-</td>
<td>YouTube</td>
</tr>
<tr>
<td>✔</td>
<td>❌</td>
<td>-</td>
<td>YouTube</td>
</tr>
<tr>
<td>❌</td>
<td>❌</td>
<td>-</td>
<td>YouTube</td>
</tr>
<tr>
<td>✔</td>
<td>/</td>
<td>Randomized payload</td>
<td>/</td>
</tr>
</tbody>
</table>
What triggers classification?

<table>
<thead>
<tr>
<th>Original IPs</th>
<th>Original Ports</th>
<th>Other changes</th>
<th>Services detected as</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>YouTube</td>
</tr>
<tr>
<td>✗</td>
<td>✓</td>
<td>-</td>
<td>YouTube</td>
</tr>
<tr>
<td>✓</td>
<td>✗</td>
<td>-</td>
<td>YouTube</td>
</tr>
<tr>
<td>✗</td>
<td>✗</td>
<td>-</td>
<td>YouTube</td>
</tr>
<tr>
<td>✓</td>
<td>✓ /</td>
<td>Randomized payload</td>
<td>/</td>
</tr>
</tbody>
</table>

Diagram: A flowchart showing the interaction between a Postman, router, packet shaper, gateway, and a replay server.
What triggers classification?

<table>
<thead>
<tr>
<th>Original IPs</th>
<th>Original Ports</th>
<th>Other changes</th>
<th>Services detected as</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Green check mark]</td>
<td>![Green check mark]</td>
<td>-</td>
<td>![YouTube]</td>
</tr>
<tr>
<td>![Red x]</td>
<td>![Green check mark]</td>
<td>-</td>
<td>![YouTube]</td>
</tr>
<tr>
<td>![Green check mark]</td>
<td>![Red x]</td>
<td>-</td>
<td>![YouTube]</td>
</tr>
<tr>
<td>![Red x]</td>
<td>![Red x]</td>
<td>-</td>
<td>![YouTube]</td>
</tr>
<tr>
<td>![Green check mark]</td>
<td>![Green check mark]</td>
<td>Randomized payload</td>
<td>![HTTP]</td>
</tr>
</tbody>
</table>
What triggers classification?

<table>
<thead>
<tr>
<th>Original IPs</th>
<th>Original Ports</th>
<th>Other changes</th>
<th>Services detected as</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>✔</td>
<td>-</td>
<td>YouTube</td>
</tr>
<tr>
<td>✗</td>
<td>✔</td>
<td>-</td>
<td>YouTube</td>
</tr>
<tr>
<td>✔</td>
<td>✗</td>
<td>-</td>
<td>YouTube</td>
</tr>
<tr>
<td>✗</td>
<td>✗</td>
<td>-</td>
<td>YouTube</td>
</tr>
<tr>
<td>✔</td>
<td>✔ / ✗</td>
<td>Randomized payload</td>
<td>HTTP /</td>
</tr>
</tbody>
</table>
### What triggers classification?

<table>
<thead>
<tr>
<th>Original IPs</th>
<th>Original Ports</th>
<th>Other changes</th>
<th>Services detected as</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>-</td>
<td>YouTube</td>
</tr>
<tr>
<td>× ✓</td>
<td>✓ ✓</td>
<td>-</td>
<td>YouTube</td>
</tr>
<tr>
<td>✓ ×</td>
<td>✓ ×</td>
<td>-</td>
<td>YouTube</td>
</tr>
<tr>
<td>× ×</td>
<td>✓ ×</td>
<td>-</td>
<td>YouTube</td>
</tr>
<tr>
<td>✓ ✓ / ×</td>
<td>✓ ✓ / ×</td>
<td>Randomized payload</td>
<td><strong>HTTP</strong> / <strong>P2P</strong></td>
</tr>
</tbody>
</table>
What triggers classification?

<table>
<thead>
<tr>
<th>Original IPs</th>
<th>Original Ports</th>
<th>Other changes</th>
<th>Services detected as</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>✔</td>
<td></td>
<td>YouTube</td>
</tr>
<tr>
<td>✔</td>
<td>✔</td>
<td></td>
<td>YouTube</td>
</tr>
<tr>
<td>✔</td>
<td>X</td>
<td></td>
<td>YouTube</td>
</tr>
<tr>
<td>✔</td>
<td>✔</td>
<td></td>
<td>YouTube</td>
</tr>
<tr>
<td>☑</td>
<td>☑ / ☒</td>
<td>Randomized payload</td>
<td>HTTP / P2P</td>
</tr>
<tr>
<td>☑</td>
<td>☑ / ☒</td>
<td>Replaced (first packet only): youtube with amktube</td>
<td>HTTP / P2P</td>
</tr>
<tr>
<td>☑</td>
<td>☑ / ☒</td>
<td>Replaced (first packet only): youtube with amktube (HOST header only)</td>
<td>YouTube / YouTube</td>
</tr>
<tr>
<td>☑</td>
<td>☑ / ☒</td>
<td>Added 100 bytes to the beginning of first packet</td>
<td>HTTP / P2P</td>
</tr>
<tr>
<td>☑</td>
<td>☑ / ☒</td>
<td>Added 100 bytes to the end of first packet</td>
<td>YouTube / YouTube</td>
</tr>
<tr>
<td>☑</td>
<td>☑ / ☒</td>
<td>Replaced GET with AMK</td>
<td>HTTP / P2P</td>
</tr>
<tr>
<td>☑</td>
<td>☑ / ☒</td>
<td>Added “GET ” to beginning of first packet</td>
<td>YouTube / YouTube</td>
</tr>
</tbody>
</table>
1. Methodology: how do we do it?
   • Record and Replay

2. Validation: does it actually work?
   • Replays maintain features of application traffic
   • Shapers classify replays correctly
   • Area test performs well in all 3 shaping regions

3. Deployment and findings
Implementation

Server
• Developed in python (~2K LOC)
• Deployed on EC2 in all regions
• Will be deployed on other VM hosts outside Amazon

Client
• Desktop client in python
• Android and iOS app (pre-loaded with traces)
Differentiation in the wild

<table>
<thead>
<tr>
<th>ISP</th>
<th>YouTube</th>
<th>Netflix</th>
<th>Spotify</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verizon</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tmobile</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sprint</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boost</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackWireless</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SimpleMobile</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NET10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Differentiation in the wild

- Tested in early 2015

<table>
<thead>
<tr>
<th>ISP</th>
<th>YouTube</th>
<th>Netflix</th>
<th>Spotify</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verizon</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Tmobile</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ATT</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Sprint</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Boost</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>BlackWireless</td>
<td>60%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>H2O</td>
<td>37%</td>
<td>45%</td>
<td>65%</td>
</tr>
<tr>
<td>SimpleMobile</td>
<td>36%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>NET10</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
</tbody>
</table>

- m: content modified on the fly
- p: translucent proxies change connection behavior

Tested in early 2015
Differentiation in the wild

- Tested in early 2015

<table>
<thead>
<tr>
<th>ISP</th>
<th>YouTube</th>
<th>Netflix</th>
<th>Spotify</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verizon</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Tmobile</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ATT</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Sprint</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Boost</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>BlackWireless</td>
<td>60%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>H2O</td>
<td>37%</td>
<td>45%</td>
<td>65%</td>
</tr>
<tr>
<td>SimpleMobile</td>
<td>36%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>NET10</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
</tbody>
</table>

- m: content modified on the fly
- p: translucent proxies change connection behavior
Differentiation in the wild

- Tested in early 2015

<table>
<thead>
<tr>
<th>ISP</th>
<th>YouTube</th>
<th>Netflix</th>
<th>Spotify</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verizon</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Tmobile</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ATT</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Sprint</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Boost</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>BlackWireless</td>
<td>60%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>H2O</td>
<td>37%</td>
<td>45%</td>
<td>65%</td>
</tr>
<tr>
<td>SimpleMobile</td>
<td>36%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>NET10</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
</tbody>
</table>

- m: content modified on the fly
- p: translucent proxies change connection behavior
Differentiation in the wild

- Tested in early 2015

<table>
<thead>
<tr>
<th>ISP</th>
<th>YouTube</th>
<th>Netflix</th>
<th>Spotify</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verizon</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Tmobile</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ATT</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Sprint</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Boost</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>BlackWireless</td>
<td>60%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>H2O</td>
<td>37%</td>
<td>45%</td>
<td>65%</td>
</tr>
<tr>
<td>SimpleMobile</td>
<td>36%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>NET10</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
</tbody>
</table>

- m: content modified on the fly
- p: translucent proxies change connection behavior

Tested in early 2015
Differentiation in the wild

• Tested in early 2015

<table>
<thead>
<tr>
<th>ISP</th>
<th>YouTube</th>
<th>Netflix</th>
<th>Spotify</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verizon</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Tmobile</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ATT</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Sprint</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Boost</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>BlackWireless</td>
<td>60%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>H2O</td>
<td>37%</td>
<td>45%</td>
<td>65%</td>
</tr>
<tr>
<td>SimpleMobile</td>
<td>36%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>NET10</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
</tbody>
</table>

• m: content modified on the fly
• p: translucent proxies change connection behavior

Not permitted in the US since July 2015
Differentiation in the wild

- Tested in early 2015
- Tested again in Aug’15

<table>
<thead>
<tr>
<th>ISP</th>
<th>YouTube</th>
<th>Netflix</th>
<th>Spotify</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verizon</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Tmobile</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ATT</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Sprint</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Boost</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>BlackWireless</td>
<td>60%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>H2O</td>
<td>37%</td>
<td>45%</td>
<td>65%</td>
</tr>
<tr>
<td>SimpleMobile</td>
<td>36%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>NET10</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
</tbody>
</table>

- m: content modified on the fly
- p: translucent proxies change connection behavior

Not permitted in the US since July 2015
Differentiation in the wild

- Tested in early 2015
- Tested again in Aug’15
  
  *No performance differences observed*

<table>
<thead>
<tr>
<th>ISP</th>
<th>YouTube</th>
<th>Netflix</th>
<th>Spotify</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verizon</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Tmobile</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ATT</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Sprint</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>Boost</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>BlackWireless</td>
<td>60%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>H2O</td>
<td>37%</td>
<td>45%</td>
<td>65%</td>
</tr>
<tr>
<td>SimpleMobile</td>
<td>36%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>NET10</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
</tbody>
</table>

- m: content modified on the fly
- p: translucent proxies change connection behavior
Differentiation in the wild

- Tested in early 2015
- Tested again in Aug'15
- No performance differences observed

ISP

- Verizon
- Tmobile
- ATT
- Sprint
- Boost
- BlackWireless
- H2O
- SimpleMobile
- NET10

<table>
<thead>
<tr>
<th>ISP</th>
<th>m</th>
<th>m</th>
<th>m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verizon</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tmobile</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sprint</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boost</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackWireless</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2O</td>
<td>37%</td>
<td>45%</td>
<td>65%</td>
</tr>
<tr>
<td>SimpleMobile</td>
<td>36%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NET10</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
</tbody>
</table>

Not permitted in the US since July 2015

• m: content modified on the fly
• p: translucent proxies change connection behavior

Thanks for reaching out to us. This isn’t something we wish to weigh in on at this time. Thanks for the opportunity.
Public website

- Data, code, summary results at http://dd.meddle.mobi
Summary

First general, accurate approach for detecting differentiation in cell networks from smartphones

• Validated technique for triggering differentiation on arbitrary apps without requiring access to source

• Identified and fixed issues with statistical techniques for identifying differentiation

• Wide-area study of differentiation in practice
  • Detected shaping (gone after new FCC rules)
  • Content/connection manipulation is pervasive
Summary

**First general, accurate approach for detecting differentiation in cell networks from smartphones**

- Validated technique for triggering differentiation on arbitrary apps without requiring access to source
- Identified and fixed issues with statistical techniques for identifying differentiation
- Wide-area study of differentiation in practice
  - Detected shaping (gone after new FCC rules)
  - Content/connection manipulation is pervasive

This work is generously supported in part by a Google Faculty Research Award