Neutral Net Neutrality
Expressing User Preferences with Network Cookies

Yiannis Yiakoumis, Sachin Katti, Nick McKeown
Stanford University
Who controls your network access?
Who controls your network access?
Who controls your network access?

What about the users?
What if we let users decide?

Neutral Network

ISP-defined Fast Lanes

User-Driven
Outline

• Why user preferences matter
• Expressing user preferences to the network
Outline

• Why user preferences matter
• Expressing user preferences to the network
1. Boost : User-driven Fast Lane
1. Boost : User-driven Fast Lane

- Deployed in ~300 homes (Google employees)
- One Boost lasts for one hour + Last Boost wins
- 44 websites in two weeks
How do people use Boost?

- “Business-related stuff should be faster (me or my partner)”
- I am stuck with a 2Mbps connection, and very often I need to dedicate all of it on a single task.
- I want my Netflix movie to go faster, not my daughter’s Netflix
How do people use Boost?
How do people use Boost?

Heavy-tail of user preferences
2. Zero-rating
Do users want zero-rating?

1000 smartphone owners, US, 18-65 year old, SurveyMonkey Audience, 08/15

- Interested: 63.0%
- Don't care: 37.0%

Survey results from 1000 smartphone owners in the US, aged 18-65, sourced from SurveyMonkey's Audience, in August 2015.
What do users really want to zero-rate?

Which application would you choose to zero-rate?

Heavy-tail of user preferences
What do users really want to zero-rate?

Which application would you choose to zero-rate?

~50

80% ignored by all programs!
User Preferences Takeaways

- Users have unique and diverse preferences
- Respect the heavy-tail
- Let users decide
Outline

• Why user preferences matter

• Expressing user preferences to the network
  – Why existing mechanisms don’t work
  – Network Cookies
Deep Packet Inspection

What is cnn.com ???

DPI is not expressive
DiffServ doesn’t work across network boundaries
“I want traffic X to get service Y!”

1. I want Netflix to be faster
2. I want Spotify zero-rated
3. I want low-latency Skype for work
“I want traffic $X$ to get service $Y$!”

1. I want Netflix to be faster
2. I want Spotify zero-rated
3. I want low-latency Skype for work

2. Communicate it to the network
3. Configure the network
How do we map traffic to a lane?
Network Cookies: a mapping abstraction

**Network Cookie**: A small piece of data users append to their traffic

1. Get *cookie* for each service
2. User appends cookies to the desired traffic
3. Network matches against them and enforces service
Network Cookies: a mapping abstraction

Cookie Descriptors: ID + Key + Metadata

Cookies: Unique, signed, use-once
Cookies & Cookie Descriptors

• Separation of data plane and control plane
  • *Descriptors: Authentication + Service Definition*
  • *Cookies: Generate and Match locally at dataplane*
• Get cookie descriptors through an out-of-band mechanism
• Insert cookies through an “agent” (browser, OS, application)
• Where to insert a cookie?
  • Anywhere we can put a few extra bits
    • HTTP header
    • TLS handshake
    • IPv6 extension header
    • ...
1. Discover Cookie Server and acquire descriptors

2. Sync descriptors with AP

3. Generate and add cookie to boosted flows

4. Match cookie, and enforces the service
Example Preferences

Everything goes to the best effort lane, apart from...
1. Dropbox and software updates → background lane
2. Google Hangouts and living room TV → fast lane
“I want traffic X to get service Y!”

1. I want Netflix to be faster
2. I want Spotify zero-rated
3. I want low-latency Skype for work

2. Communicate it to the network

3. Configure the network
Putting it all together...

1. Discover Cookie service and acquire cookie descriptors
2. Generate unique, use-once, signed *cookies*
3. Append them to desired traffic (HTTP header, TLS extension, TCP, ...)
4. *Match in network*
Network Cookies Properties
# Network Cookies Properties

**Simple & Expressive**

<table>
<thead>
<tr>
<th>Property</th>
<th>✔</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mapping abstraction</td>
<td>✔</td>
</tr>
<tr>
<td>Low transaction cost → Inclusive</td>
<td>✔</td>
</tr>
<tr>
<td>Composable/Delegetable</td>
<td>✔</td>
</tr>
</tbody>
</table>
## Network Cookies Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>✔️</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mapping abstraction</td>
<td>✔️</td>
</tr>
<tr>
<td>Low transaction cost ➔ Inclusive</td>
<td>✔️</td>
</tr>
<tr>
<td>Composable/Delegetable</td>
<td>✔️</td>
</tr>
<tr>
<td>Independent of packet header, payload, path</td>
<td>✔️</td>
</tr>
<tr>
<td>High Accuracy</td>
<td>✔️</td>
</tr>
<tr>
<td>Multiple transport mechanisms</td>
<td>✔️</td>
</tr>
<tr>
<td>Only client-network support required</td>
<td>✔️</td>
</tr>
</tbody>
</table>

**Simple & Expressive**

- Independent of packet header, payload, path
- High Accuracy
- Multiple transport mechanisms

**Deployable**

- Only client-network support required
## Network Cookies Properties

<table>
<thead>
<tr>
<th>Simple &amp; Expressive</th>
<th>Deployable</th>
<th>Respect trust</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mapping abstraction</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Low transaction cost → Inclusive</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Composable/Delegetable</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Independent of packet header, payload, path</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>High Accuracy</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Multiple transport mechanisms</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Only client-network support required</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Built-in-Authentication</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Protected from replay, spoofing</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Transparent, Revocable, Auditable</td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>
Outline

• Why user preferences matter
• Expressing user preferences to the network
• Conclusions
Conclusions & Next Steps

① A User-Driven approach is practical and beneficial
   – Evidence from real users (Boost, Zero-Rating)

① How we communicate user preferences is important!
   – Network cookies one way to do it

① Next Steps
   – Trials
   – Standardization
   – Net Neutrality Regulation
FORGET NET NEUTRALITY

yiannisy@stanford.edu

LET USERS DECIDE!!!
Thanks!
yiannisy@stanford.edu
BACKUP SLIDES
What music do users want to zero rate?
What music do users want to zero rate?

66% of user preferences not in Music Freedom (November 2015)
Cookie ➔ Network Configuration

Cookie Matching Configuration (forward+reverse flow)

Cookie Matching Configuration (forward+reverse flow)
Cookie \(\rightarrow\) Network Configuration

Cookie Matching

Configuration

Cookie \(\rightarrow\) DiffServ
Cookie $\rightarrow$ Network Configuration
Out-of-band (or SDN*)

* SDN → Software-Defined Networking

Flow tuple changes in the network

OOB doesn’t work with middleboxes (e.g. NAT)
What does a cookie look-like?

Low overhead, cannot be replayed or spoofed
Where do I get a cookie descriptor?

- Well-known server
- Discovery protocols
- Given by content provider

Out-of-band + authentication primitives
Scalability

![Graph showing scalability with varying packet sizes and throughput for different packet rates: 10 pkts/flow, 50 pkts/flow, and 100 pkts/flow.]