Smartphones vs. Laptops: Comparing Web Browsing Behavior and the Implications for Caching

Ioannis Papapanagiotou
ECE Dept.
NC State University
Raleigh, NC
ipapapa@ncsu.edu

Erich Nahum
IBM Research
Hawthorne, NY
nahum@us.ibm.com

Vasileios Pappas
IBM Research
Hawthorne, NY
vpappas@us.ibm.com

ABSTRACT

In this work we present the differences and similarities of the web browsing behavior in most common mobile platforms. We devise a novel Operating System (OS) fingerprinting methodology to distinguish different types of wireless devices (smartphone vs laptops) as well as operating system instances (iOS, Android, BlackBerry etc.). We showcase that most of the multimedia content in smartphone devices is delivered via Range-Requests, and a large portion of the video transfers are aborted. We also show that laptop devices have more intelligent browser caching capabilities. We investigate the impact of an additional browser cache, and demonstrate that a 10MB browser cache that is able to handle partial downloads in smartphones would be enough to handle the majority of the savings. Finally, we showcase that caching policies need to be amended to attain the maximum possible savings in proxy caches. Based on those optimizations the emulated proxy cache provides $10\% - 20\%$ in bandwidth savings.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations—*Network Management*

Keywords

Mobile Traffic, Network Usage, Smartphones, Caching

1. OVERVIEW

As smartphones reach the 1 billion threshold, they are already contributing more than 10% of Internet traffic [1]. Recently a number of studies have been conducted on smartphone usage [2, 3, 4, 5, 6]. Yet many aspects of smartphone traffic behavior are still not well understood. Their behavior may have implications for network design and provisioning, particularly if their traffic is significantly different from other devices, such as PCs.

This work exposes some of those differences by comparing the Web browsing behavior of smartphones with a control group, namely laptops. We do this by capturing a 3 weeks long full-packet trace in a wireless enterprise environment, composed of the traffic from over 3000 unique devices. We introduce an Operating System (OS) fingerprinting methodology based on data mining association rules. The rules are based on DHCP Request header fields, such as *Host-name*, *Vendor-Name*, *Parameter-Request* list, and the first 3-bytes of the MAC address - also called Organization Unique Identifier (OUI) - that tend to be indicative of an OS and similar across the same device types. Table 1 shows that this OS fingerprinting algorithm can identify 98.5% of the devices in the network, as compared to 84% for earlier approaches, which were based on HTTP User-Agents and the MAC OUI.

SIGMETRICS' 12, June 11–15, 2012, London, England, UK.
ACM 978-1-4503-1097-0/12/06.

		DHCP		HTTP & OUI [4]	
Device	**OS**	#	%	#	%
Laptops	All	2478	79.9	2182	73.6
	Windows	2107	67.9		
	Mac OS X	333	10.7		
	Linux	38	1.2		
Smartphones	All	534	17.2	398	13.4
	iOS	440	14.2		
	Android	54	1.7		
	BlackBerry	37	1.2		
	Windows	3	0.1		
Other	All	52	1.7	484	16.3
	Cisco VoIP	15	0.5	0	0
	Unclassified	37	1.2	65	2.2
	Unknown	-	-	419	14.1
All		3064	100.0	3064	100.0

Table 1: Distribution of Devices in the Trace

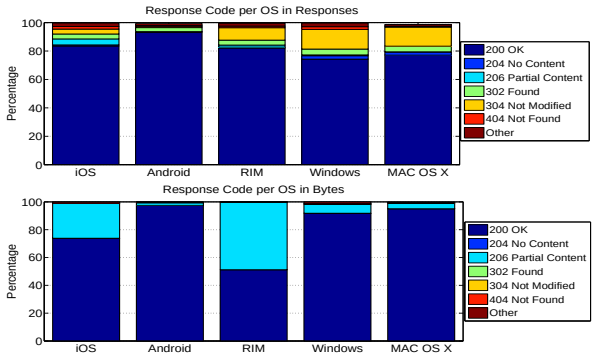

Figure 1: Response Code per Operating System

We evaluate the Web browsing behavior of the devices, as that makes up the bulk of the traffic. Figure 1 provides an overview of the response codes per device type. We observe that while in smartphones the 304 *Not Modified* is not more than 3%, in Windows and Mac OS X it is close to 20%. This code is typically returned in response to a client generating a conditional GET request, which validates a client's cached copy of a document. If a copy is valid and up-to-date in the client's cache, the server will respond with a 304 response

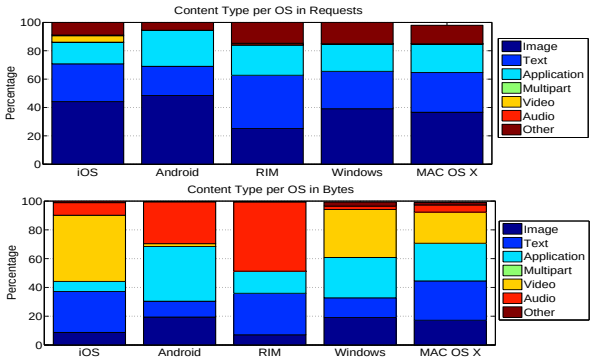

Figure 2: Content Type per Operating System

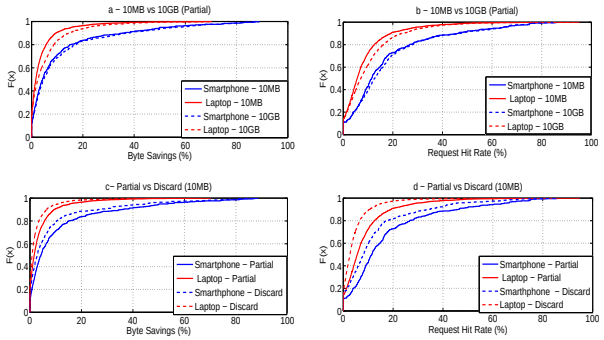

Figure 3: Comparing browser cache hit rates by varying (a-b) the size, (c-d) the storing policies.

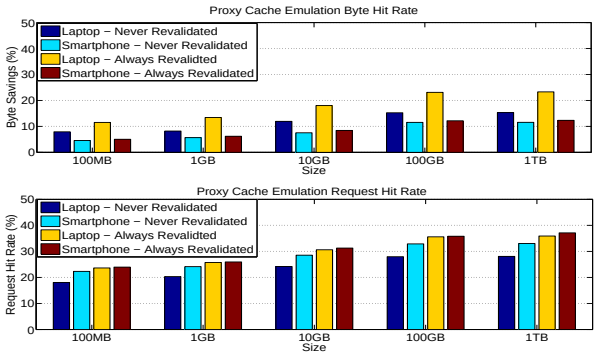

Figure 4: Proxy cache hit rate with several revalidation strategies and persistent storage sizes.

and no actual data will be transferred. Given the lack of 304 responses in the smartphones, this suggests that their browser caches are not as effective as those on the laptops.

The second observation from figure 1 is that a significant fraction of bytes downloaded by iOS and BlackBerries are returned using *206 Partial Content* responses. This status code is used when the client requests an object to be delivered in smaller chunks. For instance, such responses occur in multimedia traffic for smartphone devices, because these objects tend to be large in size. Our investigation shows that these 206 responses in iOS devices are a result of the media component *AppleCoreMedia*, which is used across different applications (we have identified more than 82 apps).

We also examine what kinds of content are downloaded by smartphones and laptops. Figure 2 shows the relative percentages of the various content types requested in the trace, both in terms of requests and in bytes. Across all platforms, a minority of requests for multimedia content (video and audio) generate the majority of the bytes transferred. Nonetheless, among the iOS requests, a higher number of video requests is associated with video objects than in other device types. In addition, 55% of the iOS bytes downloaded are multimedia, which is higher than seen on the other devices. There is also a significant portion of BlackBerry traffic that is associated with Audio. Finally, the distribution of content types on Windows and Mac OS X laptops are broadly similar.

We next turn our attention to the caching effectiveness. We study two storing policies: the *Discard* policy in which only the objects that have been downloaded in full are stored in the cache (default policy in Squid), and the *Partial* in which only the requested portion of the object is stored in the cache. We select these policies because a large portion of the objects in smartphone devices are not fully downloaded (interrupted).

In order to evaluate the browser caching we replay the trace for each device through an additional simulated cache dedicated solely to that device. If the addition of such a cache does not provide any benefits, it suggests the the browser cache is performing well. If it does provide benefits, it suggests that the browser cache is either too small or not exploiting all the available caching features of HTTP. Our results in figure 3 indicate that (a) a small browser cache size is sufficient to capture most of the savings; (b) a browser cache should be able to handle partially downloaded objects.

Finally, we evaluate the effectiveness of a Web proxy cache for our environment based on two configurations that take into account the freshness of the object in cache. In one configuration, *always revalidated*, a cached object is always assumed valid with the object at the server. In that case, the cache only revalidates the object and receives a 304 response. In the second configuration, *never revalidated*, the object is always assumed stale and the full object is retrieved via a 200 response. We show request hit rates vary from 8 to 40 percent, and bandwidth savings from 5 to 25 percent.

2. REFERENCES

[1] *Sandvine Global Internet Phenomena Report*. Spring, 2011.

[2] J. Erman, A. Gerber, K. Ramakrishnan, S. Sen, and O. Spatscheck. Over the top video: The gorilla in cellular networks. In *IMC*. ACM, 2011.

[3] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin. Diversity in smartphone usage. In *MobiSys*. ACM, 2010.

[4] A. Gember, A. Anand, and A. Akella. A comparative study of handheld and non-handheld traffic in campus wi-fi networks. In *Passive and Active Measurement*, pages 173–183. Springer, 2011.

[5] G. Maier, F. Schneider, and A. Feldmann. A first look at mobile hand-held device traffic. In *Passive and Active Measurement*. Springer, 2010.

[6] Q. Xu, J. Erman, A. Gerber, Z. Mao, J. Pang, and S. Venkataraman. Identifying diverse usage behaviors of smartphone apps. In *IMC*. ACM, 2011.