Configuring DHCP Leases in the Smartphone Era

Ioannis Papapanagiotou
ECE, 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

The Dynamic Host Configuration Protocol (DHCP) was introduced nearly 20 years ago as a mechanism for hosts to automatically acquire IP addresses. While the protocol remains the same, its usage has evolved, especially in the last decade with the introduction of mobile devices and wireless local area networks. In this paper we investigate the impact that new types of wireless devices, such as smartphones, have on DHCP. We use two one-month long traces, collected at a corporate and an educational network, and we compare side-by-side DHCP usage patterns. We develop a novel passive fingerprinting technique based on DHCP messages to determine the device type and operating system. We show that DHCP implementations vary among device types and have an effect on DHCP lease durations. To improve network address utilization, without introducing any protocol changes, we propose a new leasing strategy which takes into account device types. This strategy, compared to current approaches, improves the address utilization without considerably increasing the DHCP overhead.

Keywords

Mobile, Smartphones, DHCP, OS Fingerprinting

Categories and Subject Descriptors

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

1. INTRODUCTION

The Dynamic Host Configuration Protocol (DHCP) [8] enables devices to attach to networks without manual configuration. It does, however, require manual configuration of access policies at the DHCP servers. One of the most critical parameter of the DHCP server configuration is the lease duration, indicating how long a device can use an IP address. Setting up proper DCHP lease values has been an art rather than a science. Long lease times can lead to exhaustion of the network address pool assigned for DHCP, while short ones can result in increased broadcast traffic and unnecessary activation of wireless interfaces by power limited devices.

There have been few studies on the DHCP lease times [7, 11], both of which were done before the onslaught of smartphones in local area networks. Smartphones present a challenge in correctly configuring DHCP leases. A single device may acquire multiple IP addresses during a day due to its continuous attachment, in either asleep or active mode, with the campus-wide wireless networks. For example, as a student moves from one side of the campus to another, her devices can re-associate with various campus subnets, acquiring a different address each time. In this scenario, setting DHCP lease times even as low as one hour may not necessarily be enough to reduce network address utilization.

To further understand the impact of smartphone devices on DHCP lease times, we analyze two one-month long packet traces, collected from a corporate and an educational network. We make the following contributions:

- We develop a novel device and operating system fingerprinting technique based on DHCP messages, which significantly improves the accuracy upon previous fingerprinting techniques that are based on HTTP user-agent information [12].
- We show that DHCP message exchanges vary both across device types (e.g., laptops, smartphones) and across operating systems (e.g., iOS, Android, Windows, Mac OS X, Linux), with each device type contributing a different amount of DHCP related traffic and having a varying effect on the network address utilization.
- We propose a new DHCP leasing strategy that does not require any protocol changes, and which takes advantage of the varying usage patterns per device type. Using simulation results, driven by our traces, we show that the new strategy, compared to current approaches, improves the network address utilization sixfold without considerably increasing DHCP overhead.

2. DHCP BACKGROUND

DHCP [8] enables automatic network configuration of hosts in TCP/IP networks, with a message exchange between hosts and DHCP servers. A *discover* message is broadcasted to locate available servers. The listening server replies with an *offer*, which contains the offered IP address. The client generates a *request* ("selecting" state) asking for offered param-

IMC'12, November 14–16, 2012, Boston, Massachusetts, USA.

Trace Type	Corporate	Educational
Dates (2012)	Feb 29-Mar 25	Jan 15-Feb 15
Client MAC Addresses	2980	8726
Client IP Addresses	3435	1968
Wireless Subnets	8 * /23	/21
Leases	1h or 12h	15 min
DHCP Packets	2.16M	3.48M
TCP/UDP Bytes	2.5TB	4.9TB

Table 1: Dataset Properties

eters from one server and implicitly declining offers from all others. However, there are other occasions in which a client issues a *request* message, such as confirming correctness of previously allocated address after, e.g., system reboot ("init-reboot" state), or extending the lease on a particular network address ("renewing" or "rebinding" state[1]).

The server responds to a request with either an *acknowledgment*, if the request is granted, or a *negative acknowledgment*, in the case where the parameters are incorrect or the lease has expired. The *acknowledgment* contains the lease time for which the network address will be valid, either as a new lease, or as an update. The client may extend its lease with subsequent *request* messages sent periodically after half the lease period. For example, if the lease time is 14400 seconds, and the client is still active after 7200 seconds, it can generate a *request* message at that time to notify the server. If the lease time expires, the server assumes that the device has been disconnected from the network. The client can issue an explicit *release* message, but this is not mandated by the RFC.

Finally, an *inform* message is sent from the client to the server to ask for local configuration parameters. This only happens when the client already has an externally configured network address.

3. PACKET TRACE ANALYSIS

We capture packet traces from two large wireless local area networks, one from a corporate office network and one from an educational campus network. Table 1 summarizes the two traces, including their static lease policies.

Using the traces, we analyze the two types of data. First, we examine DHCP packets, in order to uniquely identify devices using MAC addresses, classify them based on the device type and the operating system, analyze network address utilization and emulate different lease policies. Second, we capture TCP and UDP packets using Bro 2.0 [1] to create summary log files for TCP and UDP flows, which enables us to determine the time periods in which devices are active.

3.1 Device Classification

We classify device types and operating systems by applying an *a-priori* learning algorithm [5] to generate association rules, using as input labels the following fields in the DHCP packets:

- *Host-Name*: Some devices set their host-name to a string that can identify the type of the device. For example, many iOS smartphones have names that follow the pattern of '*-iPhone', where * usually corresponds to a string related to the user.

[1]The RFC defines the "renewing" and "rebinding" as different states. Their only difference is the way they request a lease extension, i.e., broadcast or multicast.

		Corporate		Educational	
Device	**OS**	#	%	#	%
Laptop	All	2176	73.02	3970	45.50
	Windows	1787	59.97	2819	32.31
	Mac OS X	385	12.92	1131	12.96
	Linux	4	0.13	20	0.23
Smartphone	All	735	24.66	4489	51.44
	iOS	577	19.36	3069	35.17
	Android	126	4.24	1334	15.29
	BlackBerry	31	1.04	84	0.96
	Win Mobile	1	0.03	2	0.02
Other	All	69	2.32	267	3.06
	Cisco VoIP	9	0.32	-	-
	Unidentified	60	2.01	267	3.06
All		2980	100	8726	100

Table 2: Distribution of Devices in the Trace

- *Vendor-Name*: Some devices include in the vendor-name a string that can uniquely identify their operating system. For example, most versions of Microsoft Windows include the string 'MSFT'[3].
- *Parameter-Request*: Some devices generate a unique set and/or ordering of options that can be used for identification. For example, Android devices have the following options: '1 121 33 3 6 28 51 58 59'.
- *Organization Unique Identifier*: Using the IANA Ethernet assignments [2], we determine the vendor of the interface and then we identify if that vendor can be directly mapped to a specific type of device.

To quantify the confidence of the rules, we used standard data mining metrics: Support $supp(X)$ is defined as the portion of all devices that satisfy the rule x. Confidence $conf(X \Rightarrow Y)$ of an association rule $X \Rightarrow Y$ is defined as $supp(X \cap Y)/supp(X)$, where $supp(X \cap Y)$ is the support of rule $X \wedge Y$, namely, the portion of all devices that satisfy both rule X and Y. The rules that have high confidence in at least one direction ($conf(X \Rightarrow Y)$ and $conf(Y \Rightarrow X)$), and are not contradictory, are broken into their corresponding itemsets X and Y. Those rules are then used for potential classification. For example, [host-name contains 'Android'] $\Rightarrow$ [Parameter-Request-List contains '1 121 33 3 6 28 51 58 59'] happens with confidence 100%. The reverse direction [Parameter-Request-List contains '1 121 33 3 6 28 51 58 59'] $\Rightarrow$ [host-name contains 'Android'] happens with confidence 82.35%, and the remaining 17.63% are related to a device that neither has 'Android' in the host name (e.g., when the user has modified the default host-name) nor any other name from another device type. Now a host-name that contains 'Android' or a Parameter-Request-List that contains '1 121 33 3 6 28 51 58 59', can be used to classify Android devices. In other words, we assume no ground-truth but quantify every rule.

While we use an existing classification approach, we are unaware of any previous work that has used an unsupervised learning algorithm, fed with DHCP data, in order to classify devices and operating systems. Previous wireless device classification approaches were based on information in the HTTP user-agent header [12] and were able to classify up to 82% of devices. Table 2 shows the results of our approach on the two collected traces, indicating that more than 97% of the devices were classified.

Type	Corporate			Educational		
	(%)	Mean	Median	(%)	Mean	Median
iOS	51.6	251	140	35.74	158	38
Android	5.88	123	58	11.44	117	37
BlackBerry	2.68	200	48	0.88	135	42
Windows	31.16	51	24	39.6	190	45
Mac OS	7.08	52	48	12.38	148	38
Other	2.2	-	-	0.4	-	

Table 3: DHCP Requests

Corporate			
Type	**Select**	**Init-Reboot**	**Renew**
iOS	13.19	85.62	1.19
Android	72.40	17.52	10.09
BlackBerry	94.05	0.00	5.95
Windows	33.53	23.99	42.48
Mac OS X	20.33	56.18	22.49
Educational			
Type	**Select**	**Init-Reboot**	**Renew**
iOS	13.06	57.4	29.54
Android	28.46	10.78	60.76
BlackBerry	35.53	0	64.47
Windows	3.99	10.39	85.62
Mac OS X	4.91	8.79	86.3

Table 4: Relative (%) of DHCP Request Types

3.2 Lease Time Analysis

Using the DHCP-based device classification of the previous section, we analyse the DHCP request messages. Table 3 shows the acknowledged DHCP request messages[2] for each device type, as a percentage of the total requests, as well as the absolute mean and median values. We observe distinct behavioral differences between the corporate and educational network. In the corporate network, smartphones, especially iOS devices, generate considerably more DHCP requests on average as compared to laptops. In contrast, in the educational network all devices generate roughly the same number of requests. Figure 1(a), which shows the cumulative distribution of requests per device type, illustrates this more clearly. This difference between the corporate and educational network is due to the smaller lease time of the educational network, forcing all devices to generate frequent lease renewal requests, as shown by the larger number of requests per device in that network.

To better understand the differences, we present the distribution of DHCP request message types in Table 4. In the educational network, with the exception of the iOS devices, the majority of the DHCP requests are renewals. In contrast, in the corporate network a considerably smaller percentage of the requests are renewals. Given the small number of renewal requests in the corporate network, other types of requests become more prominent, revealing a number of distinctions between device types. For example, iOS devices, and to a lesser extent Mac OS X devices, generate a large proportion of init-reboot requests. In contrast, Android and BlackBerry devices generate mainly select requests, meaning once they acquire a new address, they rarely contact the DHCP server again.

This difference between Apple and other devices is attributed to the implementation of DNAv4 [4] in iOS and Mac OS X [6, 13]. DNAv4 optimizes the re-attachment to a previously connected network by attempting to reuse a previous but still valid configuration, by reducing the number of DHCP exchange messages and by using unicast ARP requests[3].

[2]We use the acknowledged requests so that we do not account for messages generated by DHCP relays.

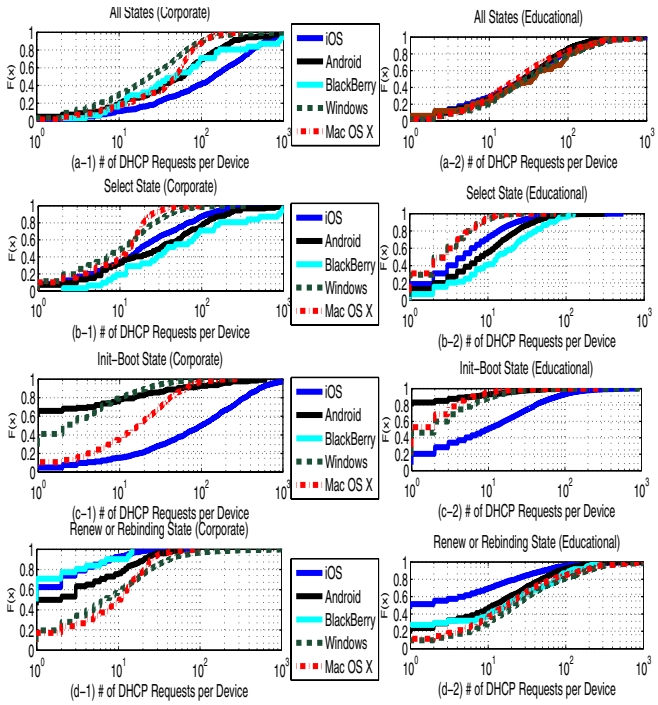

Figure 1: CDF of the number DHCP *Request* messages in the Corporate and Educational network

Finally, in Figure 2, we plot the interarrival time of the DHCP request messages. The first graph corresponds to the educational network, and the second two to the corporate network (with leases of one hour and twelve hours respectively). We also indicate with a dotted vertical line the time corresponding to half of the lease time. This is the time at which a DHCP client requests a lease extension. In the educational network, we observe that the majority of the request messages are generated at half of the lease time, indicating that they are renewal requests. In the corporate network, where the lease times are larger, the devices generate far fewer requests for renewing an IP address.

3.3 Network Access Patterns

Proper setting of DHCP lease times depends on the amount of time devices stay active in the network, i.e., the amount of time they send or receive data[4]. For a particular host, as identified by its MAC address, we define the following:

- *Active Time*: The time period, starting at the initial DHCP lease offer, up to the time a *Release* message has been issued[5] or the last time that any packet was generated, before the next lease offer.

[3]In our trace we observed unicast ARP requests associated with DHCP init-reboot requests coming from Apple devices.
[4]Note that the active time does not depend on the configuration of lease times.
[5]In both traces a *release* message is issued in $< 0.1\%$ of the leases, with an exception in Windows laptops of the corporate environment which is issued in 6.8% of the leases.

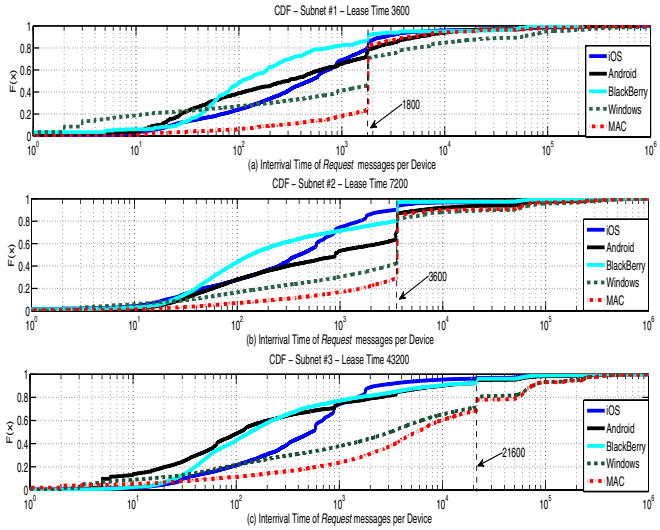

Figure 2: CDF of the interrival time of *Request* messages with different *Lease Time* setting

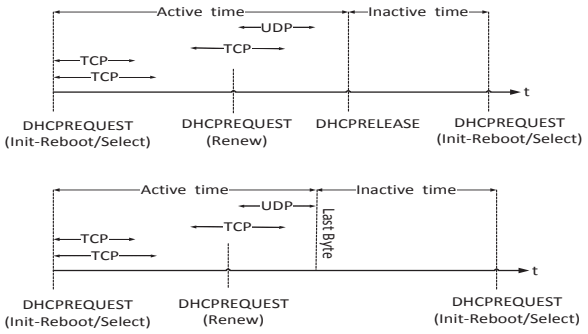

Figure 3: Calculating Active and Inactive times.

- *Inactive Time*: The time period between the end of an active period and the beginning of the next active period.

Active time starts when a device receives a DHCP acknowledgement message as a response to a DHCP request message. From the request messages, we exclude those generated when the client is either in the renewing or the rebinding state, as their purpose is to update the lease duration. However, we include the ones from the init-reboot state, where the objective is to reconfigure the leases. An illustration of active and inactive times is depicted in Figure 3.

Figure 4 shows active and inactive times for the different types of devices. We observe that smartphone active times are much smaller compared to laptop active times. We also see that active times for iOS devices are smaller than the active times of other smartphones. This happens due to a combination of reasons related to: *i)* the way users use laptops and smartphones, and *ii)* the different policies related to energy management between laptops and smartphones. User behaviour is difficult to analyze without having direct access on the devices, but we can clearly understand the effects of different energy management policies on the active

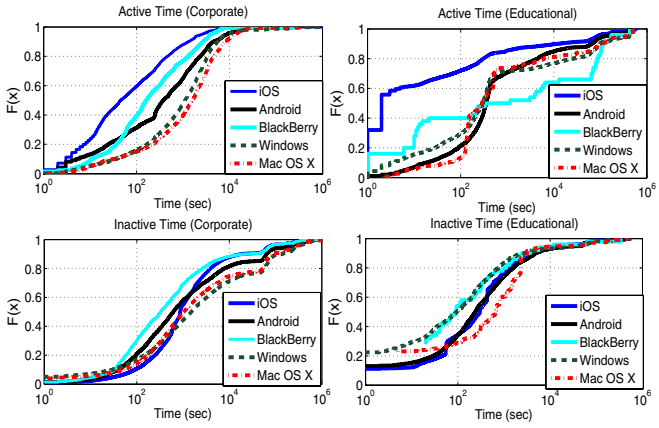

Figure 4: CDFs of Active and Inactive durations.

and inactive times. For laptops the policy has been to keep the wireless interfaces always active, while for cellphones the interface can switch off after some period of inactivity. In iOS devices, if the device is not plugged in to power and the device display is switched off, the Wi-Fi interface is also switched off and the cellular network becomes the primary interface. On the other hand, in Android devices the user is allowed to configure the WiFi sleep policy[6], although there is not a unique default policy.

Finally, it is interesting to point out that in the educational network there are a large number of smartphone devices, especially iOS devices, that have an active time of one minute or less. We attribute this to the fact that when users roam from one part of the campus to another, their devices associate with some subnets for only a brief period of time[7].

4. DHCP LEASE POLICIES

Ideally, a DHCP lease allocation policy should account for the differences in behavior of the various mobile devices. The goals of this policy should be to minimize the following, in decreasing priority:

- *Address space utilization*: The policy should use as little of the available address space as possible, in order to support the most concurrent users.
- *Broadcast traffic*: The policy should cause as few broadcasts as necessary, since broadcasts wake idle clients and consume power.
- *Server load*: The policy should minimize the load on the DHCP server, to reduce the related capex and opex expenses of running the server (including power).

We do this in two steps: first, we look at the behavior of the devices in isolation when varying the lease times over several orders of magnitude. Then, based on those behaviors, determine an approach that best meets the above goals. We wrote a simulator that, given a trace, reproduces the DHCP behavior and outputs the above metrics.

[6]There are various sleep policies: never sleep, never sleep when plugged in, sleep when screen turns off, sleep after 15 min, etc.

[7]We confirmed this with the educational network administrators, who also say that this was one of the reasons that they set the lease time to a relatively short 900 seconds.

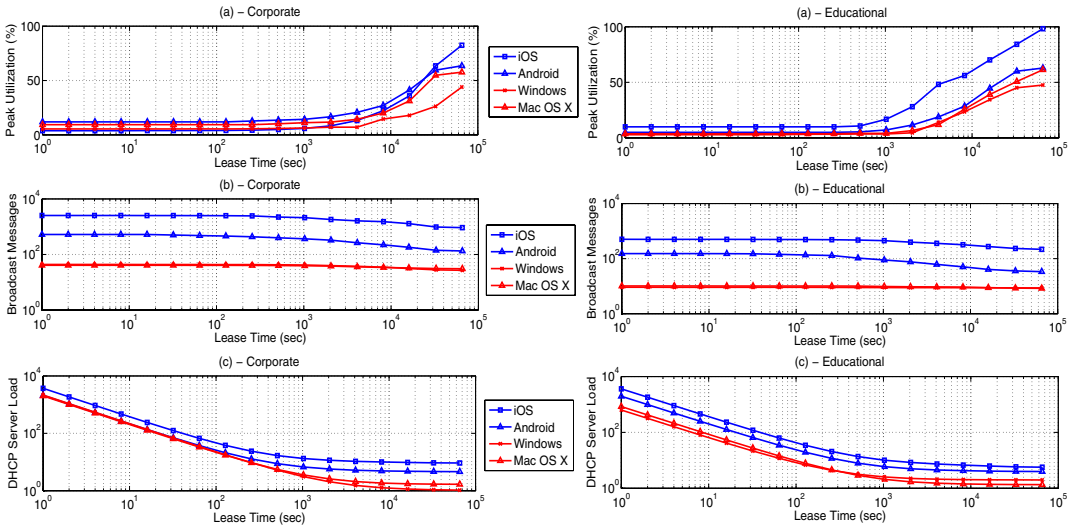

Figure 5: (a) Address space utilization (b) Broadcast messages (c) Server load, versus lease time, averaged per day and per device.

Figure 5 shows the results from our simulator. Broadly, one can see the tension between the goals in the three sets of graphs. Shorter lease times utilize the address space most efficiently, but cause large amounts of broadcast traffic and high server load. Large lease times minimize broadcasts and server load, but at the expense of poor address space utilization.

Looking more closely, in Figure 5a we see that address space utilization stays relatively flat versus lease times for each device type up until some threshold, after which utilization starts to grow logarithmically. In the corporate network and for iOS devices, the threshold is 10^3 seconds; for Androids, $2 * 10^3$ seconds, and for laptops, $4 * 10^3$ seconds. In the educational network, the same pattern holds, but with half the threshold. This is an artifact of the shorter active periods in the educational network as illustrated in Figure 4. As lease times go up, many leases are wasted on devices that have transitioned into the inactive period. This issue becomes more prevailent in smartphone devices since users have shorter access times and are more mobile compared to laptops.

In Figure 5b we observe that lease duration does not affect the number of broadcast messages generated by the laptops in both networks. Laptop users have long active times, therefore the majority of their DHCP messages are renews, which are unicast. In contrast, the number of broadcasts generated by smartphones is sensitive to the lease time. Shorter lease times incur larger numbers of broadcasts. This is because short lease time results in smartphones generating more request messages from the "selecting" state, as leases expire faster, and new leases require a full DCHP handshake, which incurs extra broadcast traffic.

In Figure 5c, we see that server load levels off at about 1,000 seconds for smartphones, but 10,000 seconds for laptops. This is due to the longer active times of the laptop users, as was shown in Figure 4.

Given these behaviors, we evaluated and compared the following DHCP lease policies:

- *Static policies*: Fixed lease times of 15 minutes, 30 minutes, and 12 hours, for all devices.
- *Exponential adaptation* [11], which allocates a short lease to a client once it arrives, and doubles the lease time every time the client renews its lease.
- *Differential lease*, which allocates different lease times based on the device type. We choose values based on our analysis of Figure 5. In the corporate network: iOS devices get 1000 seconds, Androids 2000 seconds, and Windows and Mac OS X 4000 seconds. In the educational network: iOS devices get 500 seconds, Androids 1000 seconds, and Windows and Mac OS X 2000 seconds.

Figure 6 shows the results of our simulations. We see that our differential lease policy provides a good tradeoff between our goals for both networks. It is very efficient in address utilization, almost as much as the 15 minute lease policies, yet creates less broadcast traffic and server load. Exponential adaptation, on the other hand, uses a large amount of address space, but produces low amounts of broadcast traffic and DHCP server load on a daily basis.

Different environments may have different priorities among the goals outlined above, depending on their address space size, distribution of clients (smartphones vs. laptops), etc. However, using static values requires a manual tuning process to determine the right tradeoff for the environment. Moreover, setting small static leases may have an adverse impact on the user experience. Devices have to reassociate in the wireless network when a lease expires, which in some cases can take several seconds. Our differential lease policy allows devices that tend to have longer active times to receive longer leases. Hence, it should work well across many environments, with less administrative intervention as the mixture of devices continues to change.

5. RELATED WORK

Although most wireless networks are configured to dynamically allocate IP addresses, relatively few studies exam-

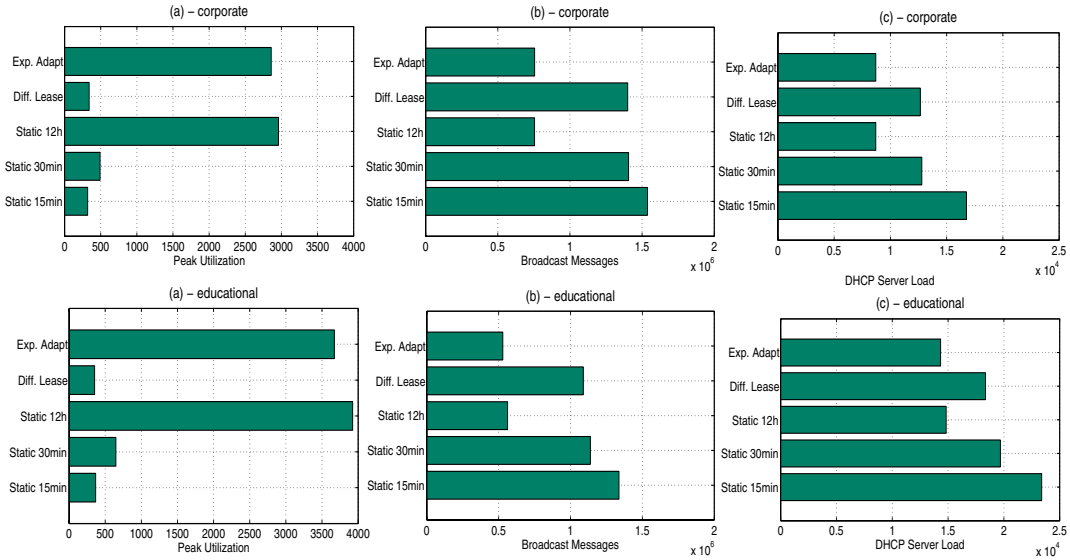

Figure 6: (a) Address space utilization (b) Broadcast messages (c) Server load, for various policies averaged per day

ine DHCP. Brick *et al.* [7] investigated the impact of lease times on DHCP performance. Khadikar *et al.* [11] studied the effects of longer DHCP lease times on address space utilization. Our work is the first to differentiate the device types and study the DHCP usage patterns of smartphones. Additionally, in contrast to previous studies, we combine DHCP and TCP/UDP behavior in order to better understand network usage patterns for each device type. Finally, our work is the first to propose DHCP leasing policies that account for the various device types and their behaviours.

With respect to device classification, Maier *et al.* [12] used a combination of IP TTL and HTTP *user-agent* information to classify device types among smartphones. Similarly, Erman *et al.* [9] identified devices based on the *user-agent* string only. Gember *et al.* [10] cross-validated the *user-agent* results with the organization unique identifier of the MAC address. In contrast, we use a different classification approach based solely on DHCP information. Our approach, in addition to being more accurate compared to the previous art, enables new DHCP leasing policies that account for different device types. Using our device classification technique, such policies can be implemented in current DHCP server software without requiring any protocol changes.

6. CONCLUSION

It has become of increasing importance for network administrators to properly allocate DHCP lease times, due to the variety of devices connected to wireless local area networks. In this paper, we show that smartphones are primary responsible for the increase in the network address utilization, and fixed lease time policies are far from optimal, even when DHCP lease times are as low as one hour. In contrast, fix leased times of 15 minutes, while they significantly decrease address utilization, produce unnecessary DHCP related overhead. To reduce this overhead, we propose a differential lease policy that assigns different lease values to each device type. The policy makes use of a novel device fingerprinting technique done at the DHCP server, without requiring any protocol changes. The main benefit of this new DHCP lease policy is that it is insensitive to the actual mixture between laptop and smartphone devices, thus removing the need to manually tune DHCP lease times as the mixture of devices continues to change.

7. REFERENCES

[1] The Bro network security monitor. `http://bro-ids.org/`.
[2] Ethernet number registration. `http://www.iana.org/assignments/ethernet-numbers`.
[3] Microsoft DHCP vendor and user classes. `http://support.microsoft.com/kb/266675`.
[4] B. Aboba, J. Carlson, and S. Cheshire. RFC 4436 - Detecting Network Attachment in IPv4 (DNAv4). IETF - `http://www.ietf.org/rfc/rfc4436.txt`, March 2006.
[5] R. Agrawal, R. Srikant, et al. Fast algorithms for mining association rules. In *VLDB*, 1994.
[6] Apple. DHCP client software. `http://www.opensource.apple.com/source/bootp/bootp-198.2/IPConfiguration.bproj/dhcp.c`.
[7] V. Brik, J. Stroik, and S. Banerjee. Debugging DHCP performance. In *Proceedings of the 7th ACM SIGCOMM Internet Measurement Conference*, pages 257–262. ACM, 2004.
[8] R. Droms. Dynamic host configuration protocol. IETF RFC, March 2007.
[9] J. Erman, A. Gerber, K. Ramakrishnan, S. Sen, and O. Spatscheck. Over the top video: The gorilla in cellular networks. In *IMC*. ACM, 2011.
[10] 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.
[11] M. Khadilkar, N. Feamster, M. Sanders, and R. Clark. Usage-based DHCP lease time optimization. In *Proceedings of the 7th ACM SIGCOMM Internet Measurement Conference*, pages 71–76. ACM, 2007.
[12] G. Maier, F. Schneider, and A. Feldmann. A first look at mobile hand-held device traffic. In *Passive and Active Measurement*. Springer, 2010.
[13] D. Simmons. Rapid DHCP redux. `http://cafbit.com/entry/rapid_dhcp_redux`.