Squeezing the Gap: An Empirical Study on DHCP Performance in a Large-scale Wireless Network

Haibo Wang, Jessie Hui Wang, Jilong Wang, Weizhen Dang, Jing'an Xue, Fenghua Li, and Jinzhe Shan

***Abstract*—Dynamic Host Configuration Protocol (DHCP) is widely used to dynamically assign IP addresses to users. However, due to little knowledge on the behavior and performance of DHCP, it is challenging to configure lease time and divide IP addresses for address pools properly in large-scale wireless networks. In this paper, we conduct the largest known measurement on the behavior and performance of DHCP in the wireless network of T University (TWLAN). We find the performance of DHCP is far from satisfactory: (1) The non-authenticated devices lead to a waste of 25% of addresses at the rush hour. (2) Address pool utilization varies greatly under the current address division strategy. (3) A device does not generate traffic for 67% of the lease time on average. Meanwhile, we observe devices of different locations and operating systems show diverse online patterns. A unified lease time setting could result in an inefficient usage of addresses. To address the problems, taking account of authentication information and online patterns, we propose a new leasing strategy. The results show it outperforms three state-of-the-art baselines and reduces the number of assigned addresses by 24% and the average total lease time by 17% without significantly increasing the DHCP server load. Besides, we further propose an adaptive address division strategy to balance the address utilization of pools, which can be deployed in parallel with the new leasing strategy and reduce the risk of address exhaustion.**

***Index Terms*—DHCP, measurement, performance, optimization**

I. Introduction

Dynamic Host Configuration Protocol (DHCP) is widely used to dynamically assign IP addresses to devices when they connect to the network [1]. In recent years, IP addresses have been almost exhausted so that available IP addresses for most enterprise networks are limited [2–4]. DHCP has the capability to manage IP addresses more efficiently and flexibly than manual configuration.

However, more and more complicated wireless network environment brings great challenges to DHCP. On one hand, roaming of devices may result in a waste of IP addresses. When a device moves from a subnet to another subnet, it will acquire multiple IP addresses [5–7], and the DHCP server cannot reclaim idle IP addresses in time. On the other hand, most wireless networks deploy authentication mechanisms, and users need to authenticate before accessing the Internet. However, many devices may be automatically associated with the wireless network without authentication. They request and occupy IP addresses, but do not generate any traffic. For example, the access points (AP) of the wireless network in T University (TWLAN) share the same Service Set Identifier (SSID). If devices ever connect to the APs, they can be associated with the wireless network and acquire IP addresses successfully without authentication, which also results in a waste of a large number of IP addresses. In locations where there are a large number of concurrent users (*e.g.*, cafeteria), even if the signal strength is very high, some devices still cannot be associated with the wireless network. One possible reason for that is IP addresses are exhausted due to the large number of non-authenticated users. Besides, when a wireless network covers a large area, in general, network administrators will configure several access controllers (AC) to manage all APs and empirically divide IP addresses into multiple address pools to associate with different ACs. An improper address division strategy will aggravate the exhaustion of addresses.

As one of the most important parameters in DHCP, the lease time determines how long a device could own an IP address. Proper lease time setting is helpful to improve the utilization of IP addresses. However, there is no clear guideline for administrators to set the lease time properly [8]. In T University, network administrators always set a fixed value based on experience. Unfortunately, setting a too long lease time will lead to a waste of IP addresses because some addresses may be occupied by the devices that already become inactive, while setting a too short lease time may greatly increase the DHCP server load. Due to lack of a comprehensive understanding on the behavior and performance of DHCP, it is challenging for network administrators to configure a proper lease time. Even worse, in large-scale wireless networks where addresses are divided into multiple pools, an improper address division strategy may result in the case that addresses are exhausted in some pools while at the same time a lot of addresses are available in other pools.

Most previous works try to improve the performance of DHCP from the perspective of finding a proper leasing strategy [8–11]. They are designed for a relatively simple network environment with fewer locations, device types and users, and they may not be able to adapt to the complicated wireless network environment. Moreover, all of the works take no account of the impact of non-authenticated users and lack a comprehensive

This work was supported by National Key Research and Development Program of China under Grant No. 2016YFB0801301, 2016YFB0801302 and 2017YFB0503703. *Corresponding author: Jessie Hui Wang (e-mail: jessiewang@tsinghua.edu.cn).*

H. Wang and W. Dang are with the Department of Computer Science and Technology, Tsinghua University and also with the Beijing National Research Center for Information Science and Technology, Beijing 100084, China.

J. H. Wang, J. Wang and F. Li are with the Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing 100084, China, and also with the Beijing National Research Center for Information Science and Technology, Beijing 100084, China.

J. Xue is with Huawei Technologies, Beijing 100085, China.

J. Shan is with the School of Engineering, Melbourne University, Parkville VIC 3052, Australia.

understanding on the behavior and performance of DHCP. To fill the gap, in this paper, we systematically conduct a large-scale measurement on the behavior and performance of DHCP in TWLAN, which has more than 59,000 individual users, 10,000 APs and 130,000 unique IP addresses. Based on the analysis, we design an effective strategy to properly set the DHCP lease time. What's more, we further propose an adaptive address division strategy to balance the address utilization of pools and reduce the risk of address exhaustion. The main contributions can be summarized as follows:

- To the best of our knowledge, we conduct the largest scale measurement on the behavior of DHCP. We find that the relationship between the trends of request messages and expiration messages varies for different locations. A mobile device ($e.g.$, mobile phone) produces more request messages than a laptop per day on average, and the number of messages produced by an IoT (Internet of Things) device falls in between. Besides, we also reveal some nonconforming and rare user behavior, which could have large impacts on the performance of DHCP.
- We present a systematic analysis on the performance of DHCP to understand the inefficiency in its operation and management. We show that there is a large gap between the number of IP addresses assigned by the DHCP server and that acquired by authenticated users. About 25% of addresses are wasted at the rush hour. Under the experience-based address division strategy, the address pool utilization for different pools varies greatly. Further, we find that a device does not generate traffic for 67% of the lease time on average. The performance of DHCP is far from satisfactory in the complicated wireless network environment. Meanwhile, there are diverse online patterns for devices of different locations and OSes. A unified lease time setting could lead to an inefficient usage of IP addresses.
- By distinguishing authenticated users and non-authenticated users, and further taking account of different online patterns of authenticated users, we propose a new leasing optimization strategy. It is light-weight and can be easily configured with parameters computed offline. We compare the proposed leasing strategy with three state-of-the-art baselines and the original leasing strategy in TWLAN on four metrics. The replay results show that our strategy outperforms other strategies and reduces the number of assigned addresses by 24% without significantly increasing the DHCP server load. Besides, the total lease time is reduced by 17% on average and the IP address utilization improves by around 10%.
- To further improve the performance of DHCP, we propose an adaptive address division strategy. It can re-divide IP addresses among different address pools according to their demanded pool sizes that are predicted using the *Random Forest Regression* model before the rush hour. Experimental results show that the adaptive address division strategy can be deployed in parallel with the new leasing optimization method and further reduce the risk of address exhaustion. The proposed method can effectively balance the address utilization of these pools, which means that each pool has available capacity to support more concurrent users.

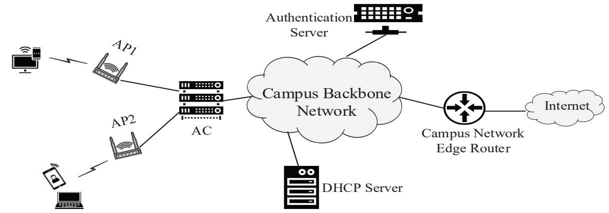


Fig. 1: The architecture of network communication in TWLAN.

The rest of the paper is organized as follows. In Section II, we introduce the datasets and methodologies used in this paper. We present our measurement results of the DHCP behavior in Section III. In Section IV, we analyze the inefficiency of DHCP. In Section V and Section VI, we respectively propose a new leasing strategy and an adaptive address division strategy to improve the DHCP performance. Finally, we summarize the related works in Section VII and conclude this paper in Section VIII.

II. Dataset and methodology

Our analysis and performance evaluation are based on several datasets collected in a large-scale campus wireless network, $i.e.$, TWLAN. TWLAN has more than 59,000 individual users, 10,000 APs and 130,000 unique IP addresses. As shown in Fig. 1, TWLAN is managed by several ACs. Each AC controls large quantities of APs in multiple buildings and these APs share the IP address pool assigned to the AC. When one device connects to the wireless network, it first initiates a DHCP request to the DHCP server to obtain one IP address. After that, it needs to communicate with the authentication server to authenticate the user's identity for accounting before accessing the Internet. After the authentication is successfully completed, all packets of the device can be forwarded to the Internet via the campus network edge router. In this paper, we mainly use the data collected in the week from May 20th, 2017 to May 26th, 2017 for measurement analysis and leasing strategy evaluation[1]. In this section, we first introduce the datasets collected in this week. Then we describe the methodologies used to find device locations and identify the types and OSes of devices, which will be used for the analysis in the following sections.

A. Dataset

In this paper, we mainly use four types of data, namely DHCP logs, authentication logs, SNMP [12] data and NetFlow [13] flow records.

(1) DHCP Logs: DHCP logs are generated by the DHCP server and record actions of the server in response to requests of DHCP clients. There are more than **22,600,000 entries** in the one-week logs. Each entry records the time of a message and IP address and MAC address of a client. Besides, it also

[1] We also use the data collected in the week from May 13th, 2017 to May 19th, 2017 to find the online patterns in Section V-B and the optimal leasing setting in Section V-C. Besides, to build the common device database in Section IV-A and train the address prediction model in Section VI, we additionally collect DHCP logs for two months.

includes some important attributes described below:
Description: It indicates the type of the DHCP message [14]. There are 8 major message types in TWLAN: *ASSIGN* (A DHCP server assigns a new IP address to a DHCP client), *RENEW* (A DHCP client updates the lease time from a DHCP server), *RELEASE* (A DHCP client releases the IP address explicitly), *EXPIRE* (A lease expires and a DHCP server reclaims the IP address), *DELETE* (A DHCP server deletes the record of an unavailable IP address), *NACK* (A lease request is denied by a DHCP server), *CONFLICT* (A DHCP server detects that an unassigned IP address has been used in TWLAN), and *EXHAUSTED* (An IP address pool has no available IP address to assign).
Host Name: It presents the host name of a DHCP client [15]. Users can customize meaningful strings as the names of their devices. It usually contains some words that can be used to identify the device type [16].
Vendor Class: It gives an identifier value that provides some clues of the device OS [15, 17]. For example, the devices with OSes of Windows series set the field as 'MSFT' by default [18], while devices of Apple usually do not set the field.

(2) Authentication Logs: In TWLAN, for authentication and accounting, an IP address must be authenticated by a user, $i.e.$, associating the address with the user, before the address can be used to access the Internet, and the user would pay for all traffic flows from and to the address. Generally speaking, the user should disassociate the address with it when it goes offline. But the user may forget to notify the authentication server. In this case, it is required that the DHCP server notifies the authentication server when it reclaims an address, otherwise the administrator may get incorrect accounting data if the IP address is assigned to other users by the DHCP server. All authentication information is recorded in authentication logs. There are more than **1,000,000 entries** in the one-week logs. Each entry indicates the identity of the authenticated user, IP address, user login time (association) and user logout time (disassociation). These entries would be used to determine whether an IP address is authenticated at any particular time point. Besides, all communication with the authentication server is based on HTTP. In this work, we capture all user authentication requests and extract the *user agent* information from HTTP request headers, which will be used to identify the types and OSes of devices.

(3) SNMP Data: SNMP is widely used for network management [19, 20]. In TWLAN, all ACs support SNMP and provide a set of data objects for administrators to monitor their status and configurations. In this work, we use a script to poll ACs every 5 minutes and get the following key-value pairs, (1) MAC address of a user device and MAC address of the AP to which it is associated, (2) MAC address of an AP and the name of the AP, and (3) MAC address of a user device and IP address it obtains by DHCP request. The SNMP data will be used to determine the location of each active device. We will discuss it later in Section II-B.

(4) NetFlow Flow Records: We deploy NFDUMP [21] to collect NetFlow flow records exported by the campus network edge router. We extract five attributes for each flow record: start time, duration, source IP address, destination IP address and total bytes of the flow. During the one week, we collect more than **3.5 terabytes** flow records in total. This dataset is used to determine whether a device is really active at any particular time point.

B. Finding Device Location

In TWLAN, each AC controls a large number of APs in different buildings, and these APs share the same IP address pool that is associated with the AC. Therefore, IP addresses in one pool can be assigned to user devices in different buildings. It is difficult to accurately determine the locations of user devices only by IP addresses.

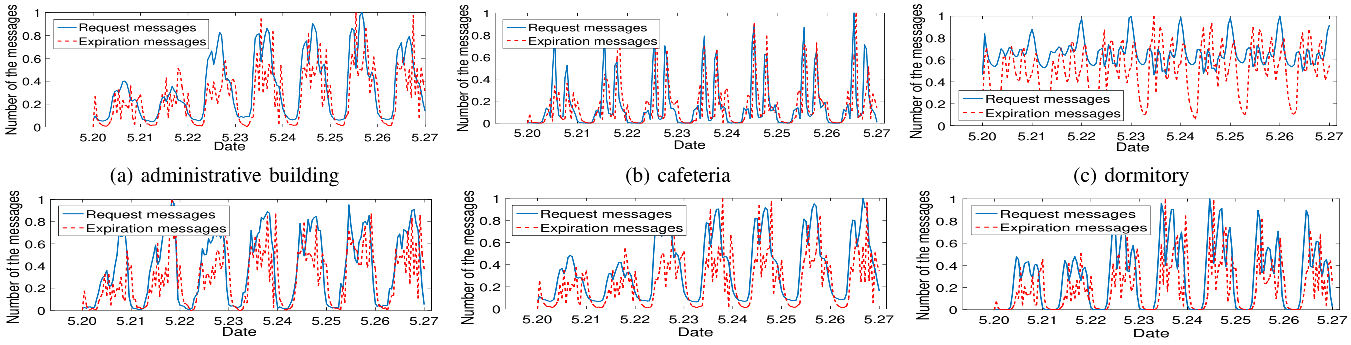
In this paper, we rely on the SNMP data to determine the location of each device at a time point. By analyzing the three types of key-value pairs collected by SNMP polling, we can get the name of the AP to which one device is connected at the time point. The APs in TWLAN are named following certain conventions. Particularly, it contains a string prefix that can determine the building where the AP is located. In this way, we can determine the locations of DHCP clients when they issue requests.

C. Identifying Device Type and OS

Previous work [8] identifies the device type and OS only by DHCP message fields. However, some users would like to modify the *Host name* field to hide the device information, which brings a great challenge for type and OS identification. The methods in the works [22, 23] identify the device type and OS based on HTTP *user agent*, which fail to deal with the cases in which *user agent* information is modified or not provided. In the work [24], the authors propose that DHCP messages and HTTP *user agent* can complement each other in the identification of device types. However, it mainly focuses on the identification of device types and the information used in the method is not sufficient to identify device OSes.

In this paper, we propose an improved method to identify device types and OSes, which combines more fields ($e.g.$, *Host Name*, *Vendor Class*, *MAC address*, etc) in DHCP logs and *user agent* in authentication logs. Our method consists of two rounds. In the first round, the fields of DHCP messages mentioned above are exploited. The fields *Vendor Class* and *oui* of *MAC address* are more credible than the field *user agent* in authentication logs because they cannot be modified by users easily, therefore we use them for the first-round identification. If there are explicit strings ($e.g.$, 'iPhone', 'MacBook', 'MSFT') in the fields (*i.e.*, *Host Name*, *Vendor Class* and *MAC address*) of DHCP logs, we can directly identify the types and OSes of the devices. For example, if *Vendor Class* is 'MSFT' for a device, we can determine that its OS belongs to the Windows series. Then we further check the field *Host name*. If there is a string 'desktop' or 'laptop' in *Host name*, we can determine that it is a Windows laptop instead of a Windows mobile device. Similarly, if there is a string 'phone', it is identified as a Windows mobile device.

Obviously, if there is no sufficient evidence in the fields of DHCP messages, the types and OSes of some devices cannot be determined definitively in the first round. For these



(a) administrative building

(b) cafeteria

(c) dormitory

(d) gymnasium

(e) research building

(f) classroom building

Fig. 2: Trends of request messages (the blue soild line) and expiration messages (the red dotted line) in different locations (values are normalized).

devices, we start the second round and conduct a keyword-based searching on the field *user agent* of their authentication logs. If it also fails to determine the types and OSes of some devices, we mark these devices as “unknown”. Even if the second round successes, we do not use its results directly because we think the information given by *user agent* is less credible. We will check if the first round provides useful information (although insufficient to draw any definitive conclusions) or clues. In case there is useful information, we will check if the identification results from the second round are consistent with the clues given by the first round. If there is no conflict between them, we can safely use the results of the second round. Otherwise, we mark the devices with conflicts as “unknown”.

We apply these three different identification methods (*i.e.*, DHCP messages based method, HTTP *user agent* based method and our proposed method) that can identify the device OSes to the one-week dataset. The results show that the types and OSes of about 98.3% of devices can be identified by our method, which greatly improves the recognition rate compared with previous methods.

During the week from May 20th, 2017 to May 26th, 2017, there are 103812 unique devices in total. We observe that Windows OS dominates laptops. The proportion is about 68.9% (#24616/#35721). Android and iOS are two major OSes used by mobile devices. Their proportions are 50.6% (#32214/#63604) and 49.1% (#31235/#63604). Besides, we also identify 557 IoT devices.

III. DHCP Behavior Measurement

In this section, we look into the dataset of DHCP messages, and try to reveal the relationship between the device behavior and the DHCP behavior. The analysis of messages of ordinary types, *i.e.*, *ASSIGN*, *RENEW* and *EXPIRE*, shows that location, device type and device OS have a clear influence on DHCP behavior, which inspires us to optimize the performance of DHCP from the views of diverse location and device OS patterns in Section V. Additionally, we also investigate messages of special types, *i.e.*, *NACK*, *CONFLICT* and *RELEASE*. These types of messages do not appear a lot, but their appearances suggest abnormal or rare user behavior. To the best of our knowledge, it is the largest scale DHCP behavior measurement in enterprise networks.

Since both message types of *ASSIGN* and *RENEW* indicate clients are requesting IP addresses, we do not differentiate them in this section, and name them as *REQUEST* messages.

A. DHCP Messages in Different Locations

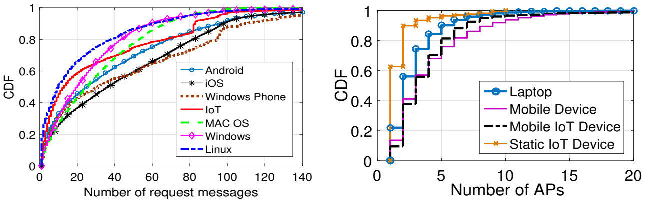
In TWLAN, *REQUEST* messages and *EXPIRE* messages together account for 80% of the total number of DHCP messages. We plot the trends of the number of these messages in different locations during the week in Fig. 2. Here, we determine the location of each DHCP message using the method described in Section II-B. Then these locations, *i.e.*, buildings, are classified into six categories: administrative building, cafeteria, dormitory, gymnasium, research building and classroom building. The classification method of buildings is inspired by previous works [19, 25].

We observe that the trends of both message types have a strong daily pattern for all location categories. In administrative buildings, research buildings and classroom buildings, the number of *REQUEST* messages and *EXPIRE* messages at weekends is significantly less than that on weekdays. Besides, it is very interesting that the trends of *REQUEST* messages are positively correlated to the trends of *EXPIRE* messages in cafeterias and gymnasiums. They approximately reach the peaks at the same time. It is because users tend to stay for a short time in these locations, and the DHCP server will assign and reclaim IP addresses frequently. However, in other locations, users tend to stay for a long time, therefore we cannot see such a positive correlation. For example, during the class time, users are likely to generate a lot of *REQUEST* messages, while a very small number of *EXPIRE* messages.

B. DHCP Messages of Different Operating Systems

In this part, we explore the DHCP behavior of different kinds of devices. Fig. 3(a) shows the CDF of the number of *REQUEST* messages generated in one day for devices with different types and OSes.

We find that a mobile device (iOS, Windows Phone, Android, *etc.*) generates more *REQUEST* messages than a laptop (Windows, MAC OS, Linux) on average. The IoT device falls in between. It can be explained by Fig. 3(b), which presents the CDF for the number of APs to which a device connects in one day. We observe that about 30% of mobile devices are associated with more than 5 APs while more than 75% of laptops are associated with less than 3 APs per day, which indicates mobile devices move more frequently than laptops. Thus, in general, mobile devices generate more *REQUEST* messages than laptops. In Fig. 3(b), we further distinguish between mobile IoT devices (*e.g.*, smart watch) and static IoT devices (*e.g.*, air humidifier) based on keywords in their names and present measurement results for them separately. We observe that more than 60% of static IoT devices are associated with only 1 AP in one day, which means that these IoT devices never move and they tend to generate less *REQUEST* messages than laptops. About 42% of mobile IoT devices are associated with more than 4 APs per day, which is similar to mobile devices. Frequent moves of these mobile IoT devices may cause that they generate a comparable number of *REQUEST* messages with mobile devices.



(a) (b)

Fig. 3: (a). CDF for the number of request messages generated by devices with different types and OSes per day. (b). CDF for the number of APs to which a device connects per day.

The other finding from Fig. 3(a) is that for the same device type, the devices of Apple generate more *REQUEST* messages than other devices per day. For example, the number of *REQUEST* messages generated by an iOS device is 40% more than that generated by an Android device on average. It can be explained by the difference of default energy management policies of different devices. The Wi-Fi interfaces of iOS devices will sleep when the device display is switched off, while the policy for Android devices can be configured and the default policy setting is "never sleep". Therefore, iOS devices will generate more *REQUEST* messages when their Wi-Fi interfaces are re-activated.

C. Special Message Types and Corresponding User Behavior

Some message types described in Section II-A do not appear frequently but reflect the non-conforming and rare user behavior, which may have large impacts on the performance of DHCP.

When a device issues a lease request that cannot be satisfied by the DHCP server, the server will deny the request and send a *NACK* message. From the one-week DHCP logs, we find there are three major reasons that may lead to the occurrence of *NACK* messages: (1) About 65% of them are due to the user mobility. When a user moves to a new network with a different address pool, it continues to request for the address it acquired in the previous lease from the previous network. In this case, the DHCP server will deny the lease request because the corresponding pool does not have the address. (2) Another reason is that a device tries to renew a lease that has expired, which accounts for about 18% of *NACK* messages. For example, the lease for the device *A* has expired and *A* tries to renew the old lease after a period of time, while the IP address has been assigned to the device *B*. Therefore, the renewal request of *A* cannot be satisfied. (3) The rest *NACK* messages are likely to be caused by misconfigurations in the network, which has been reported in [9].

A *CONFLICT* message indicates that the DHCP server detects that an unassigned IP address has been used in the network, which might be caused by non-conforming users intentionally setting static addresses for their devices instead of obtaining addresses via DHCP. Non-conforming users can infer the address pools, subnet masks and default gateways of the subnets, and further configure static IP addresses for their devices accordingly. In TWLAN, the DHCP server is configured to proactively detect whether an IP address has been used (*i.e.*, *ping* the IP address and see if there is a response) before assigning the IP address to a device. If the detection shows that the IP address has been occupied, the address would not be assigned, and the DHCP server would mark the IP address as "unusable" and generate a *CONFLICT* entry in its log. The DHCP server will periodically detect whether the address is still occupied until the answer is no. In other words, the DHCP server "loses" the address if a non-conforming user always occupies it. We name this phenomenon as *address theft*. Our measurement shows that about 155 IP addresses are stolen from the address pools by non-conforming users per day. Besides, we observe that about 40% of conflict addresses are occupied by non-conforming users for more than 3 hours.

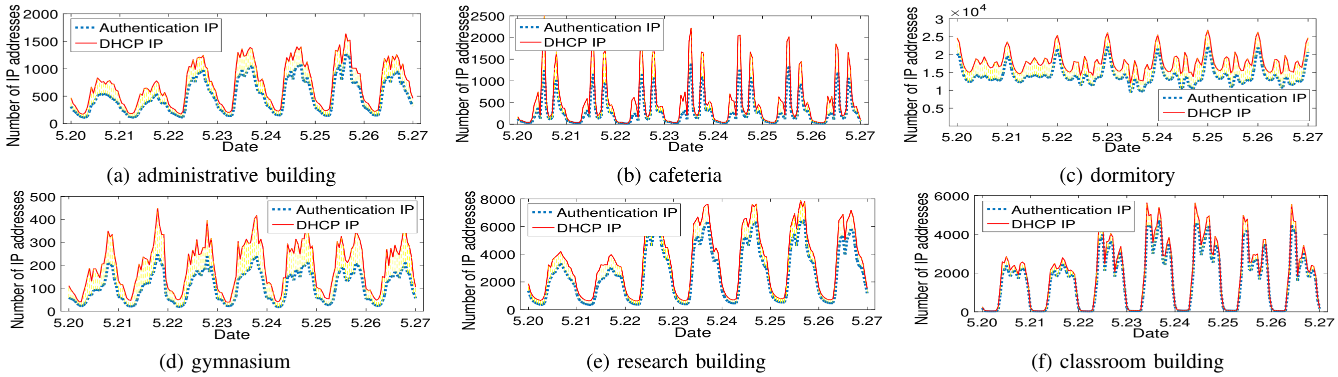
We also investigate *RELEASE* messages and explore the proportion of leases that explicitly release IP addresses during the week. We find that the proportion for each day does not vary a lot and all of them are less than 2%. It indicates users seldom release their addresses explicitly. In other words, most of idle devices will continue to occupy IP addresses until their leases expire.

IV. DHCP Inefficiency in Operation

In this section, we conduct a measurement study to understand the reasons for the inefficiency in DHCP operation and management from different perspectives. In the first part, we study the impact of non-authenticated users and address division strategy on DHCP performance. In the second part, we show the gap between the active online time of a device and its total lease time, and further explore the online patterns for different locations and device OSes. The analysis of non-authenticated users and device online patterns motivates us to propose the lease time optimization strategy in Section V, and the analysis of address pool utilization motivates us to propose the adaptive address division strategy in Section VI.

A. Non-authenticated Users and Address Pool Division

In TWLAN, only authenticated devices can access the Internet. However, non-authenticated devices can connect to



(a) administrative building

(b) cafeteria

(c) dormitory

(d) gymnasium

(e) research building

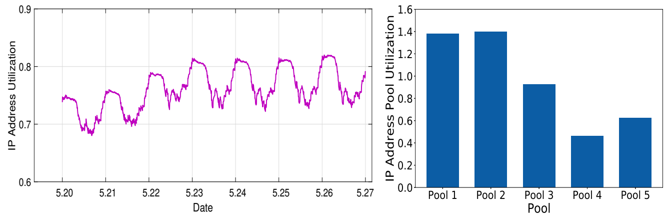
(f) classroom building

Fig. 4: The gaps between the number of IP addresses assigned by the DHCP server (the red solid line) and the number of IP addresses acquired by authenticated users (the blue dotted line) in different locations.

APs automatically if they ever connect to TWLAN. After connecting to APs, they can request addresses from the DHCP server. The DHCP server will fairly assign IP addresses to them after receiving DHCP requests. Although the devices of non-authenticated users cannot access the Internet, they occupy IP addresses. It obviously results in a waste of addresses which would otherwise be assigned to other devices.

Fig. 4 represents the differences between the number of IP addresses assigned by the DHCP server and that acquired by authenticated users during the week in different locations. We observe that the trends of the two curves are consistent. The peak time is related to the location category. For example, the number of assigned IP addresses peaks during two time periods (10:00-11:00, 14:00-16:00) in research buildings (Fig. 4(e)), which is because that most students are accustomed to researching during the periods. Students and staffs often have meals at 12:00 and 18:00, thus the number of assigned IP addresses in cafeterias (Fig. 4(b)) peaks at that time. Furthermore, we find that there exists a large gap between the two curves, which indicates a large number of IP addresses are assigned to non-authenticated users. However, the extent of the gap for different locations varies very much. In classroom buildings (Fig. 4(f)), the majority of users tend to authenticate to search for online materials. Therefore the gap is relatively small. While in administrative buildings (Fig. 4(a)) and gymnasiums (Fig. 4(d)), users prefer to do some other things rather than surf the Internet. As a result, more users would not authenticate themselves and the gaps are very large. It is interesting that the gap in dormitories is more significant than that in classroom buildings and research buildings, which can be explained by the following reason. A student usually owns multiple devices. Most of devices are more likely to appear in dormitories, such as various mobile devices and some IoT devices (*e.g.*, PlayStation, air humidifier, *etc.*). A student is less likely to use multiple devices at the same time, but these devices may be associated with the wireless network and obtain IP addresses, which leads to the relatively large gap in dormitories. While students generally study or research in classroom buildings and research buildings. They tend to only take necessary devices with them and most devices will be authenticated.

When a device issues a DHCP request for the first time, the DHCP server does not know whether the device will be authenticated later, therefore the DHCP server cannot distinguish between authenticated users and non-authenticated users and set different lease times for them. In order to solve the problem, we build a **common device database** based on the historical DHCP logs and authentication logs (data for past two months) in TWLAN. A device is regarded as a common device if its MAC address appears in DHCP logs on each weekday and at least be authenticated once in a week. We can see that common devices are the devices that are frequently used by students or staffs of T university (*i.e.*, not visitors), and these devices are more likely to be authenticated to access the Internet. The common device database is updated once a week to ensure the effectiveness. It will be used in lease time optimization in Section V.



(a)

(b)

Fig. 5: (a). The changes of the IP address utilization during the week. (b). The IP address pool utilization for 5 typical address pools in TWLAN.

As shown in Eq. (1), we define a new metric, namely IP address utilization U_{ip}, to represent the ratio of IP addresses used by common devices (IP_{common}) to all IP addresses assigned by the DHCP server (IP_{total}). The larger U_{ip} is, the less IP addresses are inefficiently used.

$$U_{ip} = \frac{IP_{common}}{IP_{total}} \tag{1}$$

We plot U_{ip} in Fig. 5(a). We observe that average U_{ip} at weekends is lower than that on weekdays, which is because that there are a lot of visitors at weekends. Besides, U_{ip} at night is higher than that in the daytime, which is because that users are almost in dormitories at night. They are more likely to be students or staffs and use their common devices. At the rush hour, we find U_{ip} is about 75%, which means about 25% of IP addresses are inefficiently used on average.

Now let us focus on the impact of the address division strategy. In TWLAN, for the convenience of management, all IP addresses are divided into multiple address pools and each pool is associated with an AC. The size of each pool is intuitively set by network administrators based on their experience and does not change over time. Obviously we would like to see the pool size is adaptive to the address demand of the AC to avoid the case that one pool is exhausted while other pools have a lot of available addresses. The intuitive setting by experience may not work well.

In order to evaluate the current address division strategy, we define IP address pool utilization U_{pool} as follows:

$$U_{pool} = \frac{IP_{demand}}{IP_{pool}} \tag{2}$$

wherein IP_{demand} is the maximum demand of IP addresses of the corresponding AC, and IP_{pool} is the size of the pool. Note that even though devices cannot get available IP addresses when address pools are exhausted, their DHCP requests are still recorded in DHCP logs in the form of *EXHAUSTED* message type. Therefore, we can accurately get the maximum demand of IP addresses during a time period. We plot Fig. 5(b) to show U_{pool} at the rush hours for five typical address pools in TWLAN (Other address pools are similar with these five cases so that we omit them for brevity). We can see that U_{pool} for both pool 1 and pool 2 exceeds 1, which means that IP addresses are exhausted for the two pools and many users cannot get addresses. While for pool 4 and pool 5, U_{pool} is very low and addresses are not fully utilized, which indicates a resource waste. U_{pool} for pool 3 is close to 1 and it faces great risk of address exhaustion. The analysis shows that U_{pool} varies greatly for different pools. The experience-based division method is not satisfactory.

B. DHCP Total Lease Time & Active Online Time

In Section III-C, we have mentioned that most user devices do not explicitly release the addresses. As a result, IP addresses cannot be reclaimed until leases expire although they have not been needed by devices. If we can predict the length of online time of the device, we can set the length of lease time accordingly to avoid the resource waste of idle devices. Obviously, a fixed length of lease time cannot achieve this goal. In this part, we collect more than 3.5 terabytes NetFlow data during the week to study the waste caused by idle devices.

We define the **active online time** in a lease as the duration from the time when a device issues a DHCP request (*ASSIGN* message) to the time when a device generates the last byte. By analyzing DHCP logs and NetFlow flow records, we can find out the active online time of all leases. DHCP **total lease time** is defined as the time period during which a device actually occupies an IP address in a lease. Fig. 6 describes the relationship between the active online time and the total lease time. Fig. 7(a) shows the CDF of DHCP total lease time and active online time for all leases. We observe that there is a large gap between the two curves and the active online time is far less than the total lease time on average. The median of the active online time is about 28 minutes, while the median of the total lease time is about 85 minutes, which means that about 67% of the time for the IP address is wasted. It aggravates the exhaustion of the limited IP addresses.

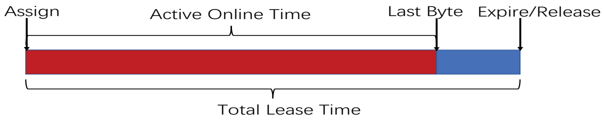


Fig. 6: Relationship between the active online time and the total lease time for a given lease.

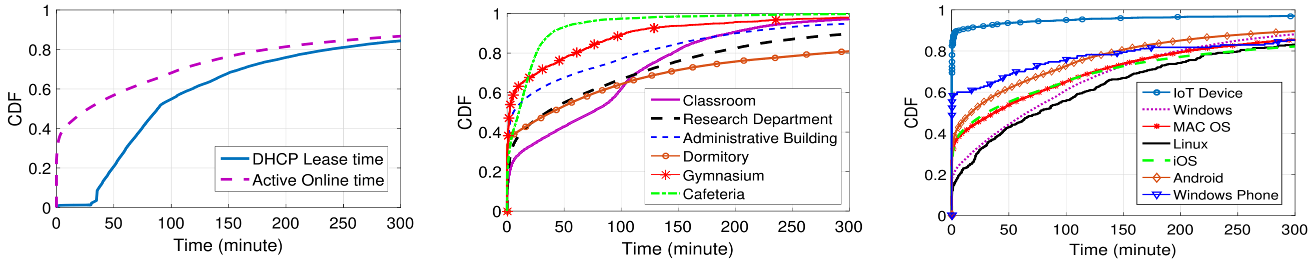
We further explore the online time length in different locations of TWLAN and plot the results in Fig. 7(b). We observe that the distribution of the online time length of devices varies for different locations. In administrative buildings, cafeterias and gymnasiums, the online time for more than 50% of devices is less than 10 minutes. In classroom buildings, the average online time is about 75 minutes and the online time for more than 80% of devices is less than 150 minutes. This is because that in general the time for a class is 95 minutes, and the time for some important classes is 155 minutes. The distributions of online time in research buildings and dormitories are similar. Both of them are long tail distributions, which can be explained by the fact that some students tend to stay in research buildings or dormitories for a long time. Fig. 7(c) shows the CDF of the active online time for devices with different OSes. Again we see the curves are different for different OSes although they are all long tail distributions. Particularly, laptops tend to have longer online time than mobile devices, and we observe that the online time for 90% of IoT devices is close to 0, which is likely to show IoT devices seldom use WiFi for communication although their WiFi modules are turned on. In summary, we can see that the locations and OSes of devices have a clear influence on the length of their active online time. A method to set DHCP lease time adaptively is necessary and the method should take account of the location and device OS.

V. Lease Time Optimization

From our measurement study, we see that the waste of IP addresses can be mitigated if we can set the lease time differently for different users and devices, *i.e.*, taking account of the authentication information, location and device OS. In this section, we introduce our ideas and implementation of the lease time optimization strategy. Moreover, we conduct experiments to demonstrate the effectiveness of our strategy by comparing with three state-of-the-art strategies and the strategy currently used in TWLAN. Particularly, we develop a replay algorithm to recover users' behavior, *e.g.*, requesting for addresses and becoming offline, and simulate DHCP message interactions as realistic as possible for performance evaluation.

A. Objective and Heuristics

The main objective is to minimize the waste of IP addresses by setting lease time properly. A shorter lease time is always helpful for our objective, but inevitably increases the DHCP server load. Therefore, the objective of the optimization is to find a good tradeoff between the efficiency and cost. In other words, our goals are to (1) *reduce the number of assigned IP*



(a) (b) (c)

Fig. 7: (a). CDF of DHCP total lease time and active online time for all devices. (b). CDF of the online time for devices in different location types. (c). CDF of the online time for devices with different OSes. (cut the long tail for readability)

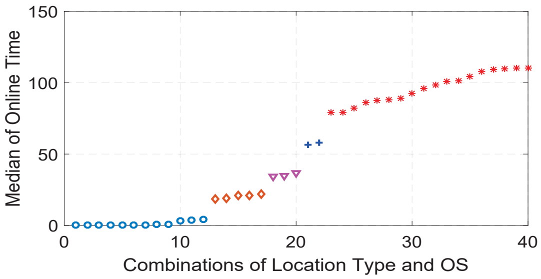


Fig. 8: The median online time for different combinations.

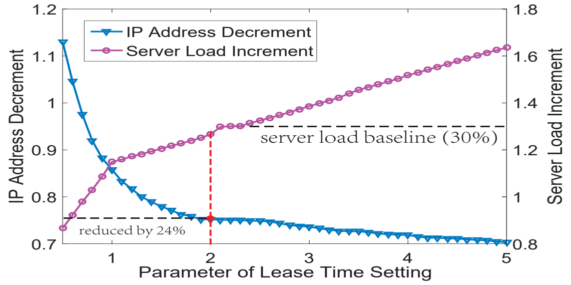


Fig. 9: The trends for the number of assigned addresses and DHCP server load with the change of the parameter x.

addresses as much as possible and (2) *restrict the increase of the DHCP server load to a threshold.*

The DHCP server load in this paper is measured by the number of received DHCP request messages and response messages. The network administrators in TWLAN suggest that the DHCP server load should not increase beyond 30%, which is determined according to the specific network environment.

We cannot predict the exact online time of each device in a lease, so we have to rely on heuristics to determine the proper lease time for each DHCP request. From our measurement and data analysis, we propose the following two heuristics:

1) For devices that are not in the common device database (built in Section IV-A), the DHCP server should assign leases with a shorter time.
2) For the device in the common device database, the longer its expected active online time (determined by its location and OS) tends to be, the longer lease time should be assigned.

The first item ensures that the DHCP server can reclaim IP addresses of devices that never authenticate in time. The second item ensures that the lease time setting takes account of the location and operating system information, and prevents a large number of DHCP messages from being generated.

B. Expected Online Time of Authenticated Users

We have found that the distribution of active online time was different for devices in different locations and with different OSes. In TWLAN, there are 6 location types and 8 OS types, which produces 48 combinations in total. Eight of them can be omitted because of their trivial proportions. We collect the data of the week from May 13th, 2017 to May 19th, 2017 and plot the CDF of the device online time for the 40 combinations respectively (For brevity, we omit the CDF figures). We find that some of them have a similar distribution, which means that they have a similar online pattern and can be treated equally.

We use the median of the online time distribution to represent the online pattern for each combination and show the median time length in Fig. 8. X-axis represents all combinations ordered by the median of the online time. We observe that these combinations of (OS, location) can be classified into five classes according to their median online time lengths. We mark these five classes by different colors and symbols. For the same class, authenticated users have an approximately same median of active online time that can be represented by one value. We define the value as *expected online time* of the class. The details are shown in Table I. The wildcard in the second column means that all locations or OSes can be matched. It follows the longest match. The expected online time will be used in lease time optimization.

C. Strategy to Set Lease Time

We propose a new strategy to set the lease time for each DHCP request as follows. After receiving a request, the DHCP

TABLE I: Five classes of authenticated users.

Index (#)	**Combinations (OS, Location)**	**Expected Online Time**
1	(MAC OS, gymnasium), (BlackBerry, *), (Windows Phone, cafeteria), (IoT, *), (Windows Phone, administrative building)	5min
2	(Android, gymnasium), (*, cafeteria)	20min
3	(Windows, cafeteria), (iOS, gymnasium), (Android, administrative building)	35min
4	(Linux, gymnasium), (iOS, administrative building)	60min
5	Other combinations	90min

server will check whether the device that issues the request is in the common device database. If it is not in the database, we set its lease time as 5 minutes. If it is a common device, the method will first obtain its location and OS. After that, the method will find its class index by matching its (OS, Location) with the entries of Table I. According to the second heuristic in Section V-A, we propose that the lease time can be set to be *linear* with the expected online time given by Table I for a common device. Let x denote the inverse of the linear coefficient, $i.e.$, the ratio of the expected online time to the lease time. It can be tuned to achieve a desired tradeoff between the DHCP server load and the maximum demand of IP addresses. A larger x will result in a heavier DHCP server load (since more *RENEW* messages are generated because of a shorter lease time). A smaller x will result in a larger demand of IP addresses (since more addresses are occupied by inactive devices because of a longer lease time).

What we need to do is to find the best value of x to make a good tradeoff in TWLAN. We gradually tune x and derive the resulting address demand and server load by replaying the DHCP logs of the week from May 13th, 2017 to May 19th, 2017 under each parameter x. The replay algorithm will be introduced in Section V-D in detail. Fig. 9 shows the trends for the number of assigned IP addresses and the DHCP server load with the change of the parameter x. X-axis represents the parameter x, left Y-axis represents the ratio of the number of assigned IP addresses in the new lease time setting to that in the original lease time setting, and right Y-axis represents the ratio of the DHCP server load in the new lease time setting to that in the original lease time setting. We observe that with the increase of the parameter x ($i.e.$, the decrease of the lease time), the benefit we get, $i.e.$, IP address decrement, becomes smaller, while the increase of the DHCP server load grows linearly. Since we need to restrict the increase of the server load to 30%, *we regard* $x = 2$ *as the sweet spot*. $x = 2$ means that the lease time should be set to a half of the expected online time. We can find it reduces the number of assigned addresses by 24% without significantly increasing the DHCP server load.

D. Replay Algorithm

We need to know the resulting IP address usage and DHCP server load under any particular lease time strategy to determine the parameter x (Section V-C) and evaluate the performance of various strategies (Section V-E). Therefore, we design a replay algorithm, which can infer the behavior of users from real-world DHCP logs (such as when they send DHCP requests to re-activate devices and when they become offline) and then replay their behavior to generate DHCP messages according to the strategy to be evaluated. With the replay algorithm, we can get the sequence of DHCP messages in case that the strategy is deployed, and then the resulting statistics for the strategy can be derived.

The replay algorithm should mainly focus on five message types ($i.e.$, *ASSIGN*, *RENEW*, *RELEASE*, *EXPIRE* and *EXHAUSTED*) because they represent a complete lease process. Among these five types, *ASSIGN* and *RELEASE* messages are directly triggered by users' actions and they would not be affected by the leasing strategy being used, therefore we can just copy these messages from original DHCP logs collected from our real-world network. Other messages would be generated at different timepoints after we deploy a new leasing strategy. In other words, they should be generated according to the strategy being used and the behavior of users recovered from original DHCP logs. Here we mainly introduce how to generate *RENEW* messages and *EXPIRE* messages

(1) Generating RENEW messages

In DHCP logs, *RENEW* messages can be caused by two different reasons. The first reason is the lease extension mechanism. At the half of the lease time, if a client is still associated with the network, it will automatically request for a lease extension, which results in a *RENEW* message. Let us name them as *lease extension RENEW* messages. The time point of generating this type of message is determined by the specific leasing strategy. Therefore, we need to ignore the *lease extension RENEW* messages in the original DHCP logs because the assigned lease time has changed, and generate new *lease extension RENEW* messages periodically based on the new leasing strategy.

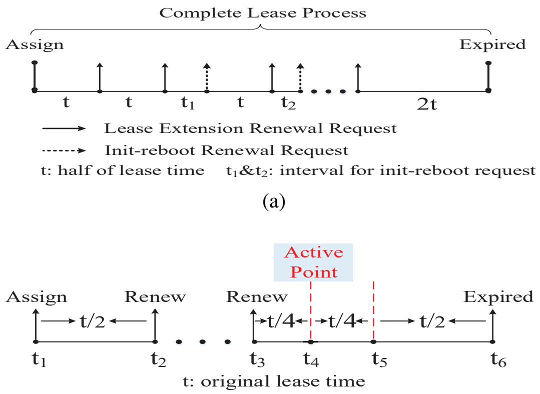
The second reason causing *RENEW* messages is user behavior, such as rebooting systems, wakening displays, and re-activating WiFi interfaces. In these cases, the device will generate a *RENEW* message to confirm the correctness of the previously obtained address. Let us name these messages as *init-reboot RENEW* messages. We need to copy these messages to the new DHCP logs because the real-world user behavior should remain the same when replaying, which means that the time to generate *init-reboot RENEW* messages under the new leasing strategy should be the same with that under the original leasing strategy.

As shown in Fig. 10(a), the arrival of init-reboot requests is stochastic while lease extension requests are always generated at the half of the lease time. We can distinguish the two kinds of requests by the time interval between two consecutive renewal requests.

(2) Generating EXPIRE messages

In the example shown in Fig. 10(b), with the original lease strategy, the lease is expired at t_6 and the last renewal request is at t_3. We can infer two results, *i.e.*, the lease time under the original lease strategy is $t = t_6 - t_3$, and the device has been disconnected from the AP at $t_5 = (t_3 + t_6)/2$ (*i.e.*, the midpoint of the original lease period), otherwise it would automatically generate a *RENEW* message to extend the lease at t_5. However, we cannot obtain the exact time point of disconnecting although we know it must be at a time point between t_3 and t_5. Therefore, we choose the midpoint of t_3 and t_5 ($i.e.$, t_4) to approximate the disconnection time point. Based on it, we can further determine the time point of the *EXPIRE* message under the new leasing strategy, which should be the time point of the last *RENEW* message under the new strategy plus the new lease time.

In fact, the replay algorithm presents a way to evaluate any method that aims to improve DHCP performance by designing leasing strategies. For a formal pseudo-code presentation of this algorithm, please refer to Appendix A.



(a)

(b)

Fig. 10: (a). Difference between *lease extension RENEW* messages and *init-reboot RENEW* messages. (b). Find the time point when the device disconnects from the wireless network in a complete lease period.

E. Performance Evaluation of Our Strategy

In previous sections, we determine parameters of our leasing strategy using the data of the week from May 13th, 2017 to May 19th, 2017. Now we try to apply the leasing strategy in subsequent weeks and evaluate its effectiveness based on the replay algorithm. For brevity, we only present the simulation results of the week from May 20th, 2017 to May 26th, 2017. For the information about more weeks, please refer to Appendix C. We employ three state-of-the-art methods as baselines to compare with our method. These state-of-the-art methods are introduced as follows:

Single Adaptation [10]: It sets a long lease time for initial lease request, and reduces the lease time for the subsequent renewal requests. The assumption behind this method is that a device is more likely to disconnect from the wireless network as its online time increases. In our experiments, we use the setting and parameters proposed by the authors in [10]. The lease time for the first request is set to 90 minutes, and for the subsequent renewal requests it is set to 30 minutes.

Exponential Adaptation [10]: It allocates a short lease time when a device first arrives, and doubles the lease time every time the device initiates a renewal request until the time setting reaches an upper bound. This method makes an assumption that a device tends to connect to the network longer if a client has been active long enough. In [10], the authors aim to reduce the server load and set the upper bound to a large value, which would result in serious resource waste. In order to be fair in our performance evaluation, we find the best parameter setting by enumeration. The upper bound is set to 60 minutes and the initial value is set to 15 minutes, which is the best to balance the address usage and server load.

OS-based Differential Lease [8]: It allocates different lease times for different device OSes to minimize the number of IP addresses at the rush hour and the DHCP server load. These two goals cannot be achieved at the same time, and the author did not present how they choose the tradeoff clearly. In the experiments, we use the same optimization objective as our strategy, *i.e.*, the DHCP server load should not increase beyond 30%. Under this condition, we choose the setting for each OS to minimize the number of assigned addresses. We plot figures to show the trends of the DHCP server load and the peak number of assigned IP addresses with the change of the lease time for different OSes, which can be referred to Appendix B. Based on them, we can get the most proper leasing settings for this strategy. For Android, iOS, Windows and MAC OS, the lease times are set to 24 minutes, 22 minutes, 28 minutes and 26 minutes respectively. For other OSes, the lease times are set to 30 minutes.

We employ the following four metrics to evaluate the performance of DHCP under different leasing strategies.

IP address usage: It refers to the number of assigned IP addresses at a time point, especially at the rush hour, which is one of the most important optimization goals.

DHCP server load: It is measured by the number of request messages and response messages received by the DHCP server.

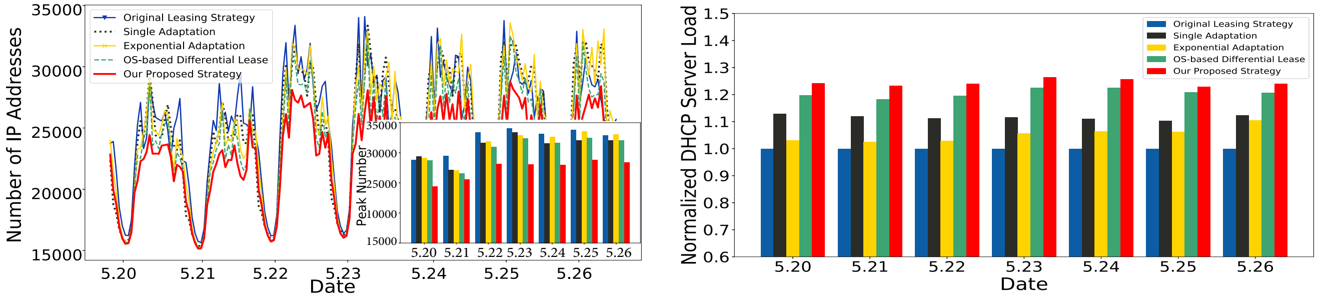
Total lease time: It is the length of the period during which a device occupies the IP address assigned by the DHCP server in a lease.

IP address utilization: It is defined in Eq. (1), and reflects the proportion of IP addresses used by common devices at a time point. In general, a high IP address utilization means that most addresses are used efficiently.

Now we compare our proposed method with the three baseline methods and the original leasing strategy with a fixed lease time in TWLAN.

Fig. 11(a) shows the number of assigned IP addresses during the week, in which the smaller inner plot represents the peak number of IP addresses at the rush hour for each day. Fig. 11(b) shows the normalized DHCP server load during the week. We observe that the number of assigned IP addresses can be reduced significantly if our proposed method is applied. At the rush hour, it saves more than 6000 IP addresses (reduced by about 24%, which is consistent with the result in Section V-C). Meanwhile, DHCP server load increases by about 25% on average, which satisfies the requirement that the server load should not increase beyond 30%. However, for *OS-based Differential Lease strategy*, although the overall IP address usage is better than that in the original leasing strategy, the trend fluctuates greatly. The peak number of assigned IP addresses is only reduced by about 10%, which is far less effective than our method. Besides, DHCP server load also increases by about 20%, which is comparable with our method. It means that the dimension of OS alone cannot accurately reflect the device online patterns in the large-scale wireless network environment. For *Single Adaptation* and *Exponential Adaptation*, although the increment of DHCP server load is not obvious, the peak number of assigned IP addresses is close to that in the original leasing strategy (even larger in some cases). It reflects that behavior characteristics vary for different users and the basic assumptions for the two leasing strategies do not necessarily apply to all devices. A good leasing strategy should differentiate diverse user behaviors. In summary, the experiment results of IP address usage and DHCP server load demonstrate that the our leasing strategy is effective.

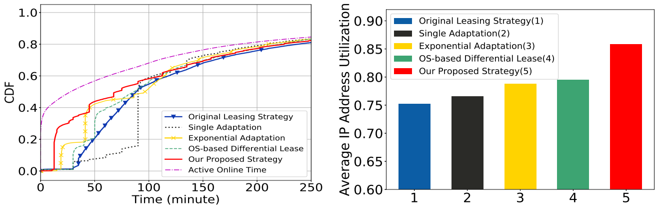
Fig. 12(a) shows the CDF of the total lease time for all devices under different leasing strategies. We find that the average total lease time is reduced by 17% in our method and



(a)

(b)

Fig. 11: (a). The trends of the number of assigned IP addresses during the week for different leasing strategies, and the smaller inner plot represents the peak number of IP addresses for each day. (b). The normalized DHCP server load during the week for different leasing strategies.



(a)

(b)

Fig. 12: (a). The CDF of DHCP total lease time for all devices in different leasing strategies. (b). IP address utilization for different leasing strategies at the rush hour.

it is more close to the active online time of a user, which means that the assigned IP addresses can be reclaimed in time, while the average total lease time under other baseline strategies is even longer than that under the original leasing strategy. Fig. 12(b) presents IP address utilization. We observe that our method achieves a significant improvement on IP address utilization compared with other baseline methods, and it is about 10% higher than the original leasing strategy. We believe it benefits from that our method carefully distinguishes between authenticated and non-authenticated users. In summary, the experiment results demonstrate that our method greatly improves the usage efficiency of assigned IP addresses.

VI. Adaptive Address Division

Although the leasing strategy presented in Section V can significantly reduce the waste of IP addresses, it is still possible that some devices cannot get addresses if the number of concurrent users exceeds the total number of IP addresses in the corresponding address pool. The analysis in Section IV-A has shown that the experience-based address division strategy currently used in TWLAN is not satisfactory. We find that there is a significant difference in the number of demanded IP addresses of each day for a given pool, which reveals that a fixed pool size may not be reasonable. Therefore, we propose an adaptive address division strategy to improve the IP address pool utilization U_{pool}.

A. The Design of Adaptive Address Division Strategy

Let us name the number of IP addresses demanded by users of an address pool as *demanded pool size*. The basic idea is to split the address space for address pools according to the demanded size of each pool. The problem is how we can predict the demanded size of a pool in a day in advance? If we can predict the demanded pool size, we can just proportionally allocate addresses to pools according to their demands at the beginning of each day.

Although the trend of the number of assigned IP addresses for each address pool has a strong daily pattern, the day-to-day peak value varies a lot, and it is difficult to predict from the historical data because the user behavior is independent and there may exist burst due to important events in some days. Fortunately, we find that the peak value of a day is likely to have a strong correlation with the number of assigned IP addresses earlier in the day. Considering that the peak value generally appears after 10:00, we expect that we can make a prediction based on the previous trend (before 10:00) and re-divide addresses to meet the real demand of each pool. From 0:00 to 6:00, there are few assigned IP addresses and the trend has no obvious change, which prevents us from extracting valuable features from the period. As a result, we aim to predict the peak value of demanded addresses by extracting features from 6:00 to 9:00 so that IP addresses can be re-divided before 10:00.

The design of the adaptive address division strategy is based on a supervised machine learning regression model. In the training phase, we continuously collect DHCP logs for two months. After that, for each pool, we extract the values of the features selected by us to form a feature vector and find the corresponding demanded pool size in each day. These data will be used to train a regression model. In the prediction phase, after feeding the newly obtained feature vector into the model, the demanded pool size can be predicted. In this work, we choose *Random Forest Regression* as the prediction model, which has been proven effective in many application scenarios, *e.g.*, WiFi performance and Internet path latency prediction [26–28]. The experiments presented in Appendix D also show that *Random Forest based* model outperforms other popular regression models.

Inspired by the previous works [26, 28], we empirically exploit 18 statistics of the curve of the number of assigned IP addresses during the period from 6:00 to 9:00 as candidate features. We name them as curve features. Besides, we observe that the trends at weekends are significantly different from those on weekdays. To improve the accuracy, we add a temporal feature to distinguish weekends from weekdays.

We further select the most effective features from these 19 candidate features based on *relative information gain* (RIG) [27], which reflects the reduction percentage of the uncertainty of the predicted target after knowing the value of a certain feature. The detailed description and the RIG values of these candidate features are presented in Appendix E. Among all candidate features, six of curve features have RIG lower than 0.01, therefore we exclude them. The other 12 curve features are selected, *i.e.*, the number of assigned IP addresses in 6:00, 7:00, 8:00 and 9:00, the average value from 6:00 to 8:00, the average value from 7:00 to 9:00, the 25th, 50th and 75th percentiles of the number of assigned IP addresses from 6:00 to 9:00, and the 50th and 75th percentiles and the maximum of curve gradient from 6:00 to 9:00 respectively. The RIG for the temporal feature is also relatively high, which means that it is valuable for the prediction model. Therefore, we retain the temporal feature.

B. Evaluation of Adaptive Address Division Strategy

In this part, we evaluate the performance of the proposed adaptive address division strategy. We first focus on the accuracy of the prediction model (*Random Forest Regression*) and adopt the *coefficient of determination* R^2 and the *relative error* as the evaluation metrics, which are respectively defined as Eq. (3) and Eq. (4),

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3}$$

$$Relative\ Error = \frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{y}_i - y_i|}{y_i} \tag{4}$$

where n represents the number of test samples, y represents the real peak values of the test samples, $\bar{y}$ represents the average value of y, and $\hat{y}$ represents the predicted values. R^2 measures how well the regression prediction values approximate the real data points. The closer R^2 is to 1, the better the prediction values fit the real data points. *Relative error* measures the significance of the prediction error. The smaller *relative error* is, the less significant the prediction errors are, and the better the prediction results are.

Fig. 13 shows the prediction results of *Random Forest Regression* on the test data. X-axis and Y-axis respectively represent the true address demand and the predicted address demand. We use the line $y = x$ as the baseline, which represents that the predicted values are exactly the same as the real values. We can see that almost all of the points are near the baseline and present a positive linear correlation. The value of R^2 is about 0.96 and the *relative error* is about 2.8%, which indicate the effectiveness of the prediction model.

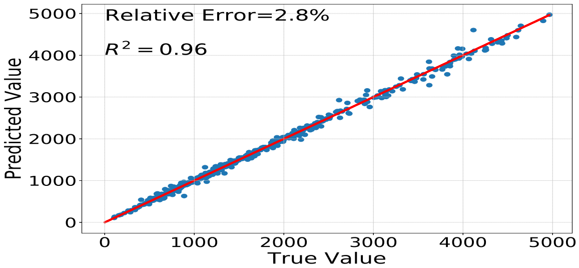
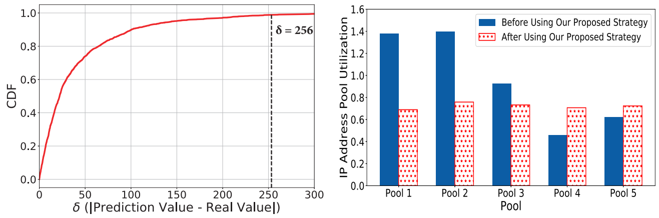


Fig. 13: The prediction results of *Random Forest Regression* on the test data.



(a) (b)

Fig. 14: (a). The distribution of the absolute value of the difference between the predicted value and the true value. (b). The results of the IP address pool utilization for the 5 typical pools after using the adaptive address division strategy and the leasing optimization strategy.

Let δ represent the absolute value of the difference between the predicted value and the true value. We plot Fig 14(a) to show the distribution of δ. We find that almost all values of δ on the test data are less than 256, which is equivalent to the size of a $/24$ address block. There are only few large values (less than 0.5%), and they are caused by unusual bursts (*e.g.*, large-scale and school-wide activities in one location), which are not representative and can be omitted. Therefore, in practice, to reduce the risk of address shortage, we use the sum of the predicted value and 256 as the demanded pool size for a pool. Furthermore, by convention, we use $/24$ as the granularity of address allocation among pools.

Finally, we simultaneously deploy both the adaptive address division strategy and the leasing optimization strategy proposed in the previous section, and evaluate the IP address pool utilization for the 5 typical address pools mentioned in Section IV-A on the data of the week from May 20th, 2017 to May 26th, 2017. As shown in Fig 14(b), compared with the original experience-based strategy, our strategy can balance the U_{pool} for all pools. We can see that the address pool utilization is around 70% for all 5 pools. At the same time, the leasing optimization strategy effectively reduces the demanded size of each pool. For example, pool 1, pool 2 and pool 3 avoid the address exhaustion problem successfully ($U_{pool} < 1$) and the address pool utilization is reduced significantly, while pool 4 and pool 5 have a larger U_{pool}, which means we can make full use of addresses. In summary, the proposed adaptive address division strategy together with the leasing optimization strategy achieves a good result that all pools have available capacity to satisfy more concurrent users.

VII. Related Work

There exists some research related to our work [8–11, 24, 29–31]. Brik *et al.* describe the potential problems of DHCP [9]. Das *et al.* propose that setting shorter lease time for handheld devices can improve the address utilization [24, 29]. Khadilkar *et al.* explore two leasing strategies to optimize IP address usage and DHCP server load [10]. Papapanagiotou *et al.* carefully study the behavior of DHCP for different device types and OSes, and further propose a leasing strategy that takes into account the differences between devices [8]. Li *et al.* design a load-aware algorithm to set lease time with the aim of reducing the DHCP overhead [11]. In [30], the authors theoretically analyze the effect of lease time setting on the address usage and the DHCP server load. Although above studies have tried to improve DHCP performance from the perspective of the lease time, optimization goals of some of them are different from ours. Furthermore, all of them are designed for simple network environments and do not take account of a lot of useful information (*e.g.*, location and authentication information), therefore they cannot be adapted to the complex network environment such as TWLAN. Besides, none of them systematically analyze the problems of the lease time setting, such as the impact of non-authenticated users on IP address utilization, and most of them do not consider the effect of the address division strategy in large-scale wireless networks. Our work fills the gaps. This paper extends our previous work [32] in the following aspects. We analyze the problem of the experience-based address division strategy and propose an adaptive division strategy to further reduce the risk of address exhaustion. We describe the design of DHCP replay algorithm in detail, and conduct more experiments to compare the proposed leasing strategy with three baselines to demonstrate the effectiveness of our strategy.

Many previous works propose different methods to identify the types and OSes of devices [8, 22–24]. Papapanagiotou *et al.* [8] present that some fields of DHCP messages can be used to identify OSes of devices. However, users are likely to modify some fields (*e.g.*, *Host Name*) to hide the device information, which brings a great challenge for identification. Some works [22, 23] identify OSes of devices by HTTP *user agent* information. However, it fails to deal with the cases in which *user agent* information is modified or not provided. Das *et al.* [24] notice that the fields in DHCP messages and the *user agent* in HTTP messages can complement each other in the identification of device types. However, they mainly focus on the identification of device types. In our work, we propose an improved method that combines more fields of DHCP logs with HTTP *user-agent* information to identify both types and OSes of devices. The results show that the new method achieves a satisfactory recognition rate.

VIII. Conclusion

To the best of our knowledge, we conduct the largest scale measurement on the behavior and performance of DHCP. Despite its wide usage, the performance of DHCP is far from satisfactory. About 25% of IP addresses are inefficiently used due to non-common devices, and the address pool utilization varies greatly across pools. Besides, 67% of the total lease time of assigned addresses is wasted on average. Non-common devices generally exist in various wireless networks, and they occupy IP addresses but do not use networks really. Large-scale wireless networks are generally divided into multiple subnets to facilitate network management, but how to split the address space among subnets is usually determined empirically. Therefore, the DHCP inefficiency reported in Section IV is a general problem in large-scale wireless networks, especially those with authentication mechanisms.

Motivated by these findings, we propose a leasing strategy and an adaptive address division strategy to improve the performance of DHCP. Experiments show that the leasing strategy can reduce the total number of assigned IP addresses by 24%, and the adaptive address division strategy can further reduce the risk of address exhaustion for each pool. In our solution, the first proposal is to distinguish non-common devices from common devices and give non-common devices shorter leases. It should be useful and effective in all wireless networks. The second proposal is to give devices different lease times according to their attributes, such as locations, types and OSes. The basic idea of this proposal should be effective for all wireless networks, but the online patterns and the influential attributes may not be the same. In case that no clear pattern is found, we can still set the same lease time for all devices and the value of the lease time is determined by the data-driven method, which should be better than the current experience-based setting. The third proposal is to split the address space according to the predicted demand of pools. The basic idea of this proposal can be effectively used in various wireless networks, but the model proposed by us to predict the demand needs to be examined case by case or type by type.

We believe that our work is a meaningful step towards a better understanding of DHCP behavior and the improvement of DHCP performance in complex wireless network environments. We are promoting the deployment of the proposed leasing strategy and the adaptive address division strategy in TWLAN. We hope the real-world deployment in production networks can provide more valuable insights towards better DHCP performance.

References

[1] R. Droms, "Dynamic host configuration protocol," 1997.

[2] "Ipv4 address report," [EB/OL], 2017, http://www.potaroo.net/tools/ipv4/.

[3] S. Zander, L. L. Andrew, G. Armitage, and G. Huston, "Estimating ipv4 address space usage with capture-recapture," in *Local Computer Networks Workshops (LCN Workshops), 2013 IEEE 38th Conference on*. IEEE, 2013, pp. 1010–1017.

[4] A. Dainotti, K. Benson, A. King, M. Kallitsis, E. Glatz, X. Dimitropoulos *et al.*, "Estimating internet address space usage through passive measurements," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 1, pp. 42–49, 2013.

[5] G. C. Moura, C. Ganán, Q. Lone, P. Poursaied, H. Asghari, and M. van Eeten, "How dynamic is the ISPs address space? towards Internet-wide DHCP churn estimation," in *IFIP Networking Conference (IFIP Networking), 2015*. IEEE, 2015, pp. 1–9.

[6] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber, "How dynamic are ip addresses?" in *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4. ACM, 2007, pp. 301–312.

[7] L. Vu, D. Turaga, and S. Parthasarathy, "Impact of DHCP churn on network characterization," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 42, no. 1. ACM, 2014, pp. 587–588.

[8] I. Papapanagiotou, E. M. Nahum, and V. Pappas, "Configuring DHCP leases in the smartphone era," in *Proceedings of the 2012 ACM conference on Internet measurement conference*. ACM, 2012, pp. 365–370.

[9] V. Brik, J. Stroik, and S. Banerjee, "Debugging DHCP performance," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. ACM, 2004, pp. 257–262.

[10] M. Khadilkar, N. Feamster, M. Sanders, and R. Clark, "Usage-based DHCP lease time optimization," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM, 2007, pp. 71–76.

[11] F. Li, X. Wang, J. Cao, R. Wang, and Y. Bi, "How DHCP Leases Meet Smart Terminals: Emulation and Modeling," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 56–68, 2018.

[12] W. Stallings, *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Addison-Wesley Longman Publishing Co., Inc., 1998.

[13] B. Claise, "Cisco systems netflow services export version 9," 2004.

[14] "Analyze DHCP Server Log Files," [EB/OL], 2017, https://technet.microsoft.com/zh-cn/library/dd183591(v=ws.10).aspx.

[15] "DHCP Tools and Options," [EB/OL], 2017, https://technet.microsoft.com/en-au/library/dd145324(v=ws.10).aspx.

[16] X. Chen, L. Lipsky, K. Suh, B. Wang, and W. Wei, "Session lengths and ip address usage of smartphones in a university campus wifi network: Characterization and analytical models," in *Performance Computing and Communications Conference (IPCCC), 2013 IEEE 32nd International*. IEEE, 2013, pp. 1–9.

[17] S. Alexander and R. Droms, "DHCP options and BOOTP Vendor Extensions," 1997.

[18] "(Microsoft) Vendor specific DHCP options explained and demystified," [EB/OL], http://www.ingmarverheij.com/microsoft-vendor-specific-dhcp-options-explained-and-demystified/, year = 2017,.

[19] K. Sui, Y. Zhao, D. Pei, and L. Zimu, "How bad are the rogues' impact on enterprise 802.11 network performance?" in *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 2015, pp. 361–369.

[20] D. Perkins and E. McGinnis, *Understanding SNMP MIBs*. Prentice-Hall, Inc., 1997.

[21] P. Haag, "Nfdump," *Available from World Wide Web: http://nfdump. sourceforge. net*, 2010.

[22] J. Erman, A. Gerber, K. Ramadrishnan, S. Sen, and O. Spatscheck, "Over the top video: the gorilla in cellular networks," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. ACM, 2011, pp. 127–136.

[23] 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*. Springer, 2011, pp. 173–183.

[24] A. K. Das, P. H. Pathak, C.-N. Chuah, and P. Mohapatra, "Characterization of wireless multi-device users," in *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*.

[25] D. Kotz and K. Essien, "Analysis of a campus-wide wireless network," *Wireless Networks*, vol. 11, no. 1-2, pp. 115–133, 2005.

[26] S. Wassermann, P. Casas, T. Cuvelier, and B. Donnet, "Netperftrace: Predicting internet path dynamics and performance with machine learning," in *Proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*. ACM, 2017, pp. 31–36.

[27] C. Pei, Y. Zhao, G. Chen, R. Tang, Y. Meng, M. Ma, K. Ling, and D. Pei, "Wifi can be the weakest link of round trip network latency in the wild," in *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE*. IEEE, 2016, pp. 1–9.

[28] M. Konte, R. Perdisci, and N. Feamster, "Aswatch: An as reputation system to expose bulletproof hosting ases," in *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4. ACM, 2015, pp. 625–638.

[29] A. K. Das, P. H. Pathak, C.-N. Chuah, and P. Mohapatra, "Characterization of wireless multidevice users," *ACM Transactions on Internet Technology (TOIT)*, vol. 16, no. 4, p. 29, 2016.

[30] T. Van Do, "An efficient solution to a retrial queue for the performability evaluation of DHCP," *Computers & Operations Research*, vol. 37, no. 7, pp. 1191–1198, 2010.

[31] X. Wei, N. C. Valler, H. V. Madhyastha, I. Neamtiu, and M. Faloutsos, "Characterizing the behavior of handheld devices and its implications," *Computer Networks*, vol. 114, pp. 1–12, 2017.

[32] H. Wang, J. Wang, W. Dang, J. Xue, and F. Li, "Squeezing the Gap: An Empirical Study on DHCP Performance in a Large-scale Wireless Network," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1628–1636.

Haibo Wang received the B.S. degree in computer science and technology from Beijing Jiaotong University in 2015. He is now a Ph.D candidate at the Department of Computer Science and Technology, Tsinghua University. His research interests include network measurement, network management and AIOps.

Jessie Hui Wang received the B.S. degree and the M.S. degree in computer science from Tsinghua University, and the Ph.D. degree in information engineering from The Chinese University of Hong Kong in 2007. She is currently an Associate Professor with Tsinghua University. Her research interests include Internet routing, cloud computing, data analysis, network measurement and Internet economics.

Jilong Wang received the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University in 2000. He is currently a Professor with Tsinghua University. His research focuses on network measurement, location-oriented network, and SDN.

Weizhen Dang received the Bachelor degree from the Department of Computer Science and Technology, Tsinghua University in 2016. He is now studying for his master at the Department of Computer Science and Technology, Tsinghua University. His research interests include network measurement, wireless network and network management.

Jing'an Xue recieved her Ph.D and B.E. in computer science and technology from Tsinghua University and Xi'an Jiaotong University. Her research interest includes network measurement and management, especially on content delivery network measurement and intelligent anomaly detection (AIOps).

Fenghua Li received the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University in 2004. He is currently a Associated Professor with Tsinghua University. His research focuses on network management, wireless network, and network architecture.

Jinzhe Shan attained the bachelor degree of Internet of Things Engineering from Beijing University of Posts and Telecommunications in 2015. He is currently pursuing the master degree from the University of Melbourne. His research focuses on network management, machine learning and natural language processing.

Appendix

A. *The Design of the Replay Algorithm*

The replay algorithm is essential for the data-driven analysis and the performance evaluation, therefore we present the design in detail in Algorithm 1. It presents a way to evaluate any method that aims to improve DHCP performance by designing leasing strategies.

When replaying DHCP logs, we mainly focus on the *RENEW* and *EXPIRE* messages because they are affected by the user behavior as well as the leasing strategy in use. As the leasing strategy changes, *RENEW* and *EXPIRE* messages would be generated at different timepoints.

(1) Generating RENEW messages in Algorithm 1

Line 11-13 are about how we deal with lease extension *RENEW* messages. The appearance of lease extension *RENEW* messages means that devices automatically request for lease extensions. Obviously, the time point to generate this kind of messages is closely related to the leasing strategy in use. If the lease time changes, these messages should be generated at different timepoints. Therefore, when replaying, we should ignore this kind of messages in the original DHCP logs, and generate new *RENEW* messages according to the new leasing strategy.

Line 14-15 and *line 20* are about how we deal with init-reboot *RENEW* messages. The appearance of init-reboot *RENEW* messages is due to the user behavior, such as rebooting systems, wakening displays, and re-activating WiFi interfaces. We need to copy this type of messages to the new DHCP logs because the real-world user behavior should remain the same when replaying, which means that the time to generate init-reboot RENEW messages under the new leasing strategy should be the same with that under the original leasing strategy.

Besides, as shown in *line 16-19*, between two successive init-reboot *RENEW* messages, we need to generate lease extension *RENEW* messages periodically based on the new leasing strategy.

(2) Generating EXPIRE messages in Algorithm 1

As shown in *line 21*, if one record in the original DHCP logs is an *EXPIRE* message, it means the end of the corresponding original lease. We show an example of a complete original lease in Fig. 15. If the lease is expired in t_6 and the last renewal request is initiated at t_3. We can infer that the device has been disconnected from the AP at t_5 ($i.e.$, the sum of t_3 and the half of the original lease time), otherwise it would automatically generate a *RENEW* message to extend the lease. Therefore, we choose the midpoint of t_3 and t_5 ($i.e.$, t_4) to approximate the exact timepoint of disconnecting during the period from t_3 and t_5. The interval between t_4 and t_3 is one quarter of the original lease time. When replaying, in *line 23*, we first get the approximate timepoint of disconnecting by adding the initiated time of the last *RENEW* message and one quarter of the original lease time. After that, as shown in *line 24-26*, we need to generate lease extension *RENEW* messages periodically ($i.e.$, half of the new lease time) based on the new leasing strategy before the timepoint of disconnecting. Finally, we will generate the *EXPIRE* messages accordingly when the new leases expire (*line 27*).

Algorithm 1 Replay Algorithm for Leases in DHCP Log

```
Input: Original lease $L$, Parameter $x$
Output: New lease $newLease$
 1: $newTime$ = $L.leaseTime/x$;
 2: if $L.device$ not in $Common\ Device\ Database$ then
 3:     $newTime = 5$;
 4: else
 5:     $newTime =$ getLeaseTime($L.location$, $L.OS$)$/x$;
 6: Initialize a new lease $newLease$;
 7: for $i = 1$ to $L.length$ do
 8:     if $L[i].description ==$ ASSIGN or RELEASE then
 9:         append $L[i]$ to $newLease$;
10:     if $L[i].description ==$ RENEW then
11:         if $L[i]$ is not init-reboot request then
12:             // The case of lease extension request
13:             continue;
14:         else
15:             // The case of init-reboot request
16:             $t =$ time for last entry in $newLease$;
17:             while $t + newTime/2 < L[i].time$ do
18:                 $t = t + newTime/2$;
19:                 append RENEW at $t$ to $newLease$;
20:             append $L[i]$ to $newLease$;
21:     if $L[i].description ==$ EXPIRE then
22:         $t =$ time for last entry in $newLease$;
23:         $tmp = L[i-1].time + L.leaseTime/4$;
24:         while $t + newTime/2 < tmp$ do
25:             $t = t + newTime/2$;
26:             append RENEW at $t$ to newLease;
27:         append EXPIRE at $t + newTime$ to $newLease$;
28:     if $L[i].description ==$ EXHAUSTED then
29:         if There exist available IP addresses in pool then
30:             append a complete lease to $newLease$;
31:             break;
32:         else
33:             append $L[i]$ to $newLease$;
34:     if $L[i].description ==$ Others then
35:         append $L[i]$ to $newLease$;
36: return $newLease$
```

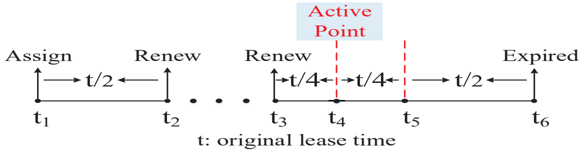
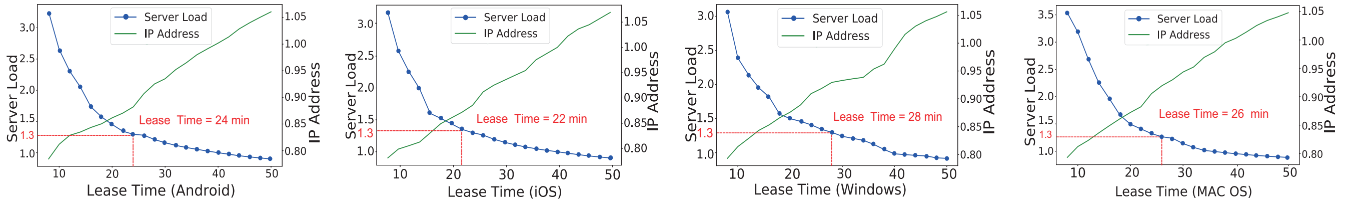


Fig. 15: Find the time point when a device disconnects from the wireless network in a complete lease.

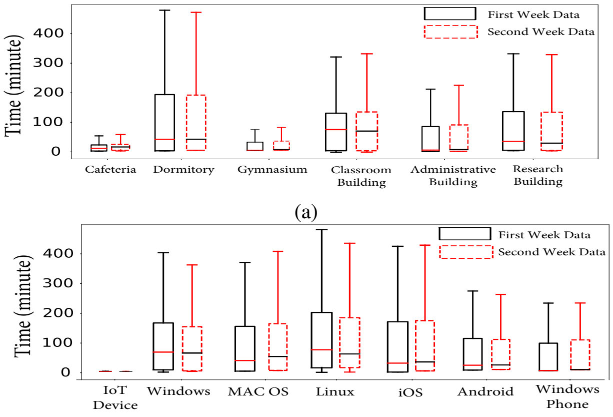
B. *Lease time settings for the baseline strategy*

The *OS-based Differential Lease* strategy allocates different lease times for different device OSes to minimize the number of IP addresses at the rush hour and the DHCP server load. To determine the optimal lease time settings for the strategy, we plot Fig. 16 to show the trends of the DHCP server load and the peak number of assigned IP addresses with the change of the lease time for different OSes. From it, we can see that for Android, iOS, Windows and MAC OS, the lease times should be set to 24 minutes, 22 minutes, 28 minutes and 26 minutes respectively. For other OSes, the lease times are set



(a) Android (b) iOS (c) Windows (d) MAC OS

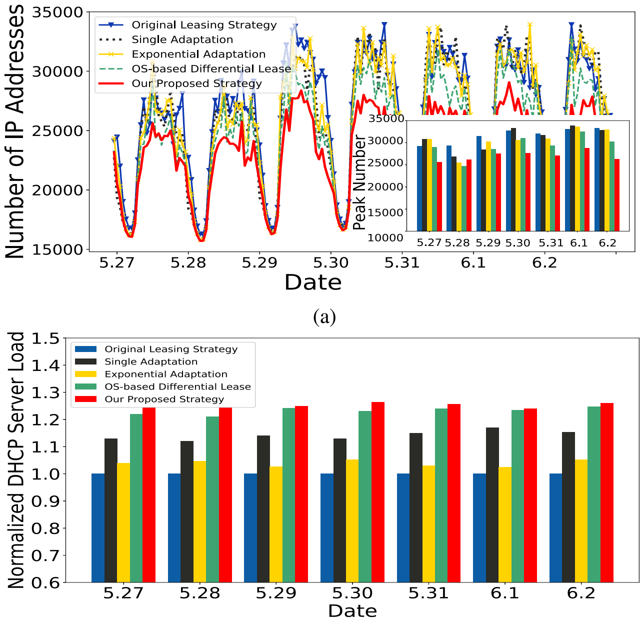
Fig. 16: Find the most proper leasing settings for different types of OSes (Android: 24min; iOS: 22min; Windows: 28min; MAC OS: 26min). Figures for other OSes are omitted because of their small proportions and similar trends. The lease times for them are set to 30min.



(a)

(b)

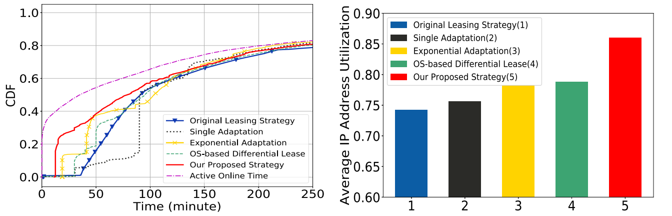
Fig. 17: (a). Comparison of the online pattern of devices of different locations for two weeks. (b). Comparison of the online pattern of devices of different OSes for two weeks.



(a)

(b)

Fig. 18: (a). The trends of the number of assigned IP addresses during the new week. (b). The normalized DHCP server load during the new week.



(a) (b)

Fig. 19: (a). The CDF of DHCP total lease time for all devices in the new week. (b). IP address utilization in the new week.

to 30 minutes. We omit the figures for them because of their small proportions and similar trends.

C. Experiments on the data of Another Week

For brevity, in the paper we only present the measurement results on the DHCP inefficiency issue and the performance evaluation of our solution based on the data of one week. Here, we would like to present more results to validate our observations in the paper. We will show the online pattern for devices of different locations and OSes is similar across different weeks. Besides, we further use the data of another week to demonstrate the similar DHCP inefficiency behavior and validate the effectiveness of the proposed leasing strategy.

(1) Similar online pattern across different weeks

We make boxplots to compare the active online time distributions of devices of different locations and OSes in the week from May 20th, 2017 to May 26th, 2017 with that in the week from May 27th, 2017 to June 2th, 2017. As shown in Fig. 17(a) and Fig. 17(b), we can see that there is no obvious change across the two weeks, which means that the behavior of user devices shows a weekly pattern and the data from one week should be sufficient to characterize the user behavior of this campus network.

(2) Similar DHCP inefficiency and effectiveness evaluation of the new leasing strategy

In our work, we report the inefficiency of the original leasing strategy in TWLAN and the effectiveness of our proposed leasing strategy based on the data of the week from May 20th, 2017 to May 26th, 2017. Here we further show the measurement and evaluation results using the data of another week from May 27th, 2017 to June 2th, 2017. Note that when we replay the DHCP logs of the new week, the setting of our proposed leasing strategy remains the same as that used for the evaluation in the paper, which is determined by the data of the week from May 13th, 2017 to May 19th, 2017 in Section V-C.

The results confirm that the main conclusions in the paper also hold for our experiments on the data of the new week. Here, we present figures and summarize the new results as follows.

- *The proposed leasing strategy can make a desired tradeoff between the DHCP server load and the peak number of assigned addresses during the new week.* Fig. 18(a) shows the number of assigned IP addresses and Fig. 18(b) shows the normalized DHCP server load during the new week. We observe that the experimental results are consistent with that in the previous week presented in Section V-E. The peak number of assigned IP addresses under our proposed leasing strategy is reduced significantly and the DHCP server load never increases beyond 30%.
- *During the new week, the active online time of devices is far less than the total lease time on average, and the proposed leasing strategy can effectively reduce the gap.* The CDF of the total lease time for all devices in the new week is shown in Fig. 19(a). We can see that in the original leasing strategy, there is still a large gap between the total lease time and the active time, which means that a large number of IP addresses are inefficiently used in the new week. The conclusion is consistent with that presented in Section IV-B. Furthermore, we find that the proposed leasing strategy still can effectively reduce the average total lease time and outperforms other strategies.
- *During the new week, non-common devices lead to a great waste of IP addresses, and the proposed leasing strategy can effectively improve the address utilization.* We show the IP address utilization in Fig. 19(b). We can see that in the original leasing strategy, about 26% of IP addresses are assigned to non-common devices, which means that the negative effect of the authentication mechanism on DHCP performance is still serious and the proportion is very close to that reported in Section IV-A. What's more, the new leasing strategy can improve it by about 12%.

The above observations show that our conclusions are valid across different weeks. Besides, it is worth mentioning that our method is data-driven. To ensure the effectiveness of the leasing strategy, the parameters ($i.e.$, common device database, online patterns of authenticated users and parameter x) of the leasing strategy can be updated periodically.

D. Performance Comparison for Different Regression Models

In Section VI-A, we choose the *Random Forest Regression* model to predict the peak value of demanded addresses for each pool. In order to further demonstrate the superiority of *Random Forest Regression*, we compare it with 6 popular regression prediction methods ($i.e.$, *Linear Regression, Logistic Regression, Bayesian Regression, Decision Tree Regression, SVM Regression, Gradient Boosting Regression*). The results of 10-fold cross validation are presented in Fig. 20. We find that *Random Forest Regression* outperforms other regression algorithms on R^2, which means that the *Random Forest based* model is more effective in this problem.

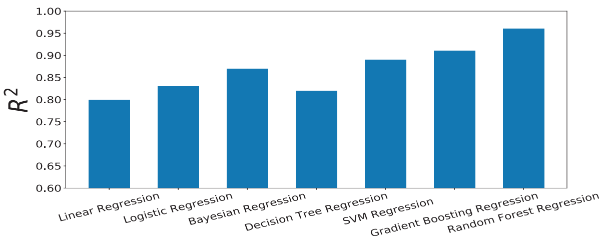


Fig. 20: Comparison between *Random Forest Regression* and the other 6 popular regression algorithms.

TABLE II: Relative information gain for all features.

Category	Index	Feature	RIG
Curve Features (only retain top 12 features)	1	#(assigned IP addresses in 6:00)	0.08
	2	#(assigned IP addresses in 7:00)	0.11
	3	#(assigned IP addresses in 8:00)	0.19
	4	#(assigned IP addresses in 9:00)	0.16
	5	average value from 6:00 to 8:00	0.08
	6	average value from 7:00 to 9:00	0.26
	7	25th percentile of #(IP addresses) from 6:00 to 9:00	0.09
	8	50th percentile of #(IP addresses) from 6:00 to 9:00	0.12
	9	75th percentile of #(IP addresses) from 6:00 to 9:00	0.15
	10	50th percentile of curve gradient	0.09
	11	75th percentile of curve gradient	0.16
	12	maximum of curve gradient	0.21
	13	minimum of curve gradient	$<$ 0.01
	14	25th percentile of curve gradient	
	15	curve gradient in 6:00	
	16	curve gradient in 7:00	
	17	curve gradient in 8:00	
	18	curve gradient in 9:00	
Temporal Feature	19	weekends or weekdays	0.15

E. Feature Selection for the Prediction Model

In order to predict the peak value of demanded addresses for each pool based on the *Random Forest Regression model*, we empirically extract *18 curve features* from 6:00 to 9:00 to represent the changes of the number of assigned IP addresses and the growth rate, and *1 temporal feature* to distinguish weekends from weekdays. The detailed description of these 19 candidate features is shown in Table II. Then we adopt *relative information gain* (RIG) to select the most significant features. RIG can reflect the reduction percentage of the uncertainty of the predicted target after knowing the value of a certain feature ($i.e.$, the contribution to the prediction task). The values of RIG for the 19 candidate features are presented in the fourth column in Table II. We ignore the candidate features in the rows from the 13th to 18th because of their low RIG values (lower than 0.01), and the top 12 curve features are selected to be used in our prediction model. We also notice that the average number of assigned IP addresses from 7:00 to 9:00 and the maximum of curve gradient contribute the most to this prediction task, and the temporal feature is also very valuable for prediction.