

Knocking Cells: Latency-based Identification of IPv6 Cellular Addresses on the Internet

Ming Wang*, Yahui Li*[‡], Han Zhang*^{†‡}, Anlun Hong*, Jun He*, Jilong Wang*[†]

*Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing, China

[†]Zhongguancun Laboratory, Beijing, China

Emails: {m-wang20, hongal20, he-j21}@mails.tsinghua.edu.cn,
liyahui_ego@126.com, zhhan@tsinghua.edu.cn, wjl@cernet.edu.cn

Abstract—IPv6 mobile networks are becoming increasingly important. Many jobs rely on understanding IPv6 mobile networks at the IP level. Previous works on cellular identification suffer from coarse identification granularity, proprietary data, or not working for IPv6. The high latency in mobile networks makes identifying cellular addresses possible using Round-Trip Time (RTT) difference. However, due to the impact of packet loss on the measurement of RTT difference, identifying cellular addresses with less overhead is challenging. In this paper, by triggering the non-zero RTT difference of the cellular /48 subnets with probes, we propose an accurate latency-based method to identify cellular /48 subnets from fixed subnets. Experiments demonstrate that the method can identify cellular subnets with a precision of 93.52% to 99.95% and a recall of 99.96% on a worldwide dataset. The overhead of measuring RTT difference reduces to at least 1/10th compared to the existing methods while robust to packet loss.

Index Terms—IPv6, mobile networks, Round-Trip Time, cellular address, identification

I. INTRODUCTION

Mobile networks are increasingly important as the infrastructure that provides wireless access to mobile users, IoT devices, and more. With the rapid development of 5G and IoT, mobile networks are rapidly moving to IPv6 [1]. Significant operators worldwide have deployed IPv6 in their mobile networks [2].

Understanding the IPv6 mobile networks at the IP level benefits lots of existing works, such as analyzing the mobile traffic [3], learning the spatial characterization of IPv6 addresses [4], [5], scanning IPv6 mobile network [7], discovering IPv6 active addresses [6] and the topology of mobile networks [8], etc.

Mobile users are always resident in mixed networks [3], and it is difficult to distinguish cellular users from fixed-line users. Previous works on identifying mobile networks have several limitations. The identification granularity of [9] is AS, which cannot further identify cellular IP addresses. The reverse DNS (rDNS) based approach [10] works for IPv4 but not for IPv6. Rula et al. [3] identify cellular IPv4 /24 and IPv6 /48 subnets using the data collected by CDN. But the data is difficult for general researchers to access. There still lacks a simple but effective method to identify IPv6 cellular addresses for the community.

Influenced by the Radio Resource Controller (RRC) state transition [12], the latency in mobile networks is high [11].

[‡] Corresponding authors

The RRC, which is specific to mobile networks and does not exist in fixed networks (e.g., home broadband network), makes it possible to use Round-Trip Time (RTT) obtained by active measurement to identify mobile networks from fixed networks [10]. The length of paths to different destinations affects RTT. The RTT difference is used to reduce the impact of network paths and reflect variations in latency within the mobile network [10]. However, due to the packet loss, multiple packets must be sent to the same target at a fixed interval to measure the RTT difference. For example, existing works [10], [11] use ten pings with an interval of one second to obtain RTT differences. The challenge in identifying IPv6 cellular addresses with RTT is to reduce the probing overhead when packet loss occurs.

This paper presents an accurate latency-based approach to identify IPv6 cellular /48 subnets and demonstrate its efficiency in reducing the overhead of active measurement. First, we show the variation of RTTs for different RRC states in mobile networks. We then describe how the transition of RRC states can be triggered to obtain a non-zero RRT difference by sending probe packets. We measure RTT differences for packets from different devices in the same /48 subnet rather than packets from the same device to significantly reduce the probing overhead. Finally, based on the characteristics of the RTT difference in cellular and fixed /48 subnets, we train a machine learning model to identify cellular /48 subnets. We validate the model on a global dataset with a precision of 93.52%~99.95% and a recall of 99.96%.

In summary, the main contributions of this paper are:

- Based on the characteristics of the RTT difference of /48 subnets, we propose an accurate latency-based method to identify IPv6 cellular subnets and validate it on a global dataset with a precision of 93.52% ~99.95% and recall of 99.96%.
- For each IP in the /48 subnet, the overhead to measure the RTT difference dramatically reduces to 1/10 compared to existing methods; When packet loss occurs, only 100 RTTs in a /48 subnet are required to identify the cellular subnet with a precision of 92.17%~99.78% and recall of 95.75%.

The rest of the paper is organized as follows: In Section II, background and related work is presented. In Section III, we analyze the downlink latency of the mobile network. The cellular /48 subnets identification method is described in Section IV and

evaluated in Section V. We discuss the ethical considerations and limitation in Section VI and VII separately. Finally, Section VIII concludes the paper and describes future work.

II. BACKGROUND AND RELATED WORK

A. IPv6 Network Periphery

IPv6 is the latest version of the Internet Protocol (IP). The IPv6 address has 128 bits and consists of two parts: the subnet prefix and the Interface ID (IID), each with 64 bits. The subnet prefix is used to identify the network and for routing purposes. The IID is used to identify the interface on the local link of the target network.

IPv6 network periphery is the last-hop routed infrastructure device connecting end-hosts or only enabling connectivity for itself [7]. Most IPv6 peripherals are private devices behind NAT in IPv4, but public devices on the Internet in IPv6, such as home routers and smartphones.

The end-to-end characterize of IPv6 exposes the peripherals to the Internet [13] and with the help of high-speed network device discovery techniques and tools [7], [14], it is possible to finding active IPv6 network peripherals on the Internet fastly. The full IPv6 address of a periphery can be found by a packet whose destination IP consists of the /64 prefix assigned to the periphery and a random 64bits IID [7], [13], [14]. In recent years, the technology has been widely used to quickly discover IPv6 addresses on CPEs (Customer Premise Equipment) in residential network [14], [15] or smartphones in mobile networks [7], [13].

B. Long Term Evolution (LTE)

We introduce the components and the RRC of mobile networks through LTE, the most widely used mobile communication system.

1) *Components of LTE*: The LTE mainly consists of User Equipment (UE), Radio Access Network (RAN), and Packet Core Network (CN), shown in Fig.1.

UE. UE is a mobile device used by mobile subscribers and equipped with integrated circuit Subscriber Identification Module (SIM) physical smart cards.

RAN. RAN is the first network access point from the UE to the Internet, which wirelessly connects the UE to the CN. The main element in the RAN is the base station, called eNodeB(Evolved Node B, eNB), which controls the radio resource between eNodeB and UE through RRC.

CN. CN is responsible for managing the mobility of the users by interacting with the RAN, for delivering Internet data connections requested by the users with the help of nodes shown in Fig.1. Serving Gateway (SGW) and Packet Data Network Gateway (PGW) are the user plane gateway nodes that route the IP traffic between the UE and the external networks. The Mobility Management Entity (MME) is the control plane node that handles mobility management, such as noticing inactive users when data comes from the Internet.

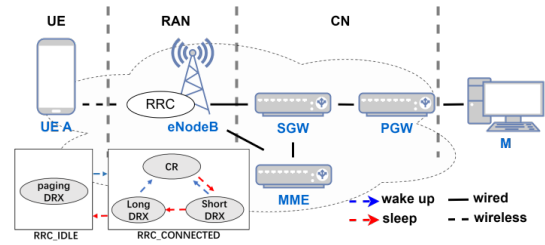


Fig. 1. The packets from the Internet reach UE through the RAN and CN in LTE. The mobile network manages the connection between UE and CN through the RRC.

2) *RRC of LTE*: To balance user experience and radio resource utilization, UE always goes into “sleep” to reduce power consumption when no user data is transmitted for a long period. Once user data comes, the mobile network wakes up the “sleeping” UE through a series of procedures with the help of the RAN and CN, such as Paging [24], Service Request Procedure [25], etc. Thus, although the UE goes into “sleep”, it is “always on” in terms of Internet access.

The key point to achieve the transition is the RRC. Mobile networks use the RRC state to manage the connection between CN and UE. As shown in Fig.1, the RRC has two states (RRC_CONNECTED and RRC_IDLE) and multiple sub-states (continuous reception (CR), short Discontinuous Reception (DRX), long DRX, and paging DRX). When in CR state, UE is allocated all the resources needed for sending and receiving user data. DRX is adopted for the “micro-sleep” of UE to reduce power consumption while providing connectivity. When in the DRX state, UE wakes up and rests periodically to save energy. Different DRXs have similar wake-up mechanisms but different wake-up cycles.

As shown in Fig.1, the RRC states transit from one to another depending on user data reception. When no data comes, the RRC state will sequentially transit to ShortDRX, LongDRX, and PagingDRX according to the parameters of eNB [16]. Once the data comes from the Internet, the RRC state of UE will transit to CR directly, regardless of its state.

While the RRC state transition mechanism is very flexible and ensures that the UE can switch between idle and active to save power, the cost is that it introduces additional delays in the downlink latency of user data [11], [12].

C. Mobile Networks Identification

It is challenging to tell whether a particular IP address comes from a cellular or fixed-line user. AS-level identification [9] is too coarse to identify the cellular IP addresses in mixed AS [3]. Rula et al. [3] proposed using logs from CDN to identify cellular /48 subnets. However, the logs are difficult to access for general researchers. In addition, the Network Information API is a non-standard Javascript API not supported by several popular mobile browsers, such as Firefox for Android and Safari for iOS. Lee et al. [10] proposed rDNS-based IPv4 addresses clustering and used RTT difference to identify the IPv4 cellular

clusters. But rDNS is rarely used by routers and UEs in IPv6 mobile networks [8], making the clustering infeasible to IPv6. Wang et al. [17] proposed identifying Cellular-IoT with the PSM (Power Save Mode) using active measurement. Due to the traffic hijacking required, it is difficult to use on a large scale.

III. USER DOWNLINK DATA LATENCY IN LTE

In this section, we analyze the latency components of user downlink data in different RRC states. We show that packets to UEs in different RRC states differ in RTTs, resulting in a non-zero RTT difference.

A. Latency Component of User DownLink Data

Fig.2 illustrates the latency spent during the packet transmission from the host M on the Internet to the UE A in the mobile network. We will use the figure to illustrate the latency introduced in different RRC states.

1) *CR state*: When the packet P is transmitted from M to UE A in the CR state, the transmission process is as follows:

- (i) The P arrives at PGW from M, with the latency T_0 ;
- (ii) The PGW sends P to SGW, with T_1 .
- (iii) The SGW sends P to eNodeB with T_3 ;
- (iv) The eNodeB forwards P to UE A, with T_4 ;
- (v) UE A reply the P to M in the opposite direction;

So, when the UE is in the CR state, the RTT of the P from M to UE A is $T_{CR} = (T_0 + T_1 + T_3 + T_4) \times 2$.

2) *IDLE state*: As the IDLE UE A releases its RRC connection to the SGW and goes to sleep to save power, the packet P delivered to it will be cached in the SGW to wait for UE A to be wakened up. First, the SGW, with the help of the MME, informs UE A to wake up and establishes the RRC connection to the SGW. When the connection is established, the UE enters the CR state. The SGW will then send P to the eNodeB serving the UE and the eNodeB will forward it to the UE. The latency that SGW cached the packet P is T_2 (blue dashed line shown in Fig.2);

So, in addition to the packet latency transmitted directly to the UE in the CR state, the RTT of the IDLE state introduces cache latency in SGW. The RTT is $T_{IDLE} = (T_0 + T_1 + T_3 + T_4) \times 2 + T_2$.

3) *DRX state*: When the packet P is delivered to UE A in the DRX state, the packet P is no longer cached in the SGW as the RRC connection between the UE and the SGW has already been established. The P will be cached in the eNodeB to wait for the UE to wake up periodically. When UE A wakes up, the eNodeB forwards P to it. The latency P cached in eNodeB is T_{eNB} (blue dashed line shown in Fig.2).

So, the RTT of the downlink latency in the DRX state is $T_{DRX} = (T_0 + T_1 + T_3 + T_4) \times 2 + T_{eNB}$.

B. The RTT Difference between Different RRC States

We summarize the RTTs of user downlink data as:

$$T = (T_0 + T_1 + T_3 + T_4) \times 2 + T_{cache} \quad (1)$$

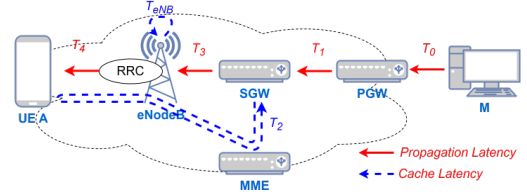


Fig. 2. Latency component of downlink data in the mobile networks.

where $T_{cache} \in \{0, T_{eNB}, T_2\}$, $0 < T_{eNB} < T_2$ in general. We can conclude that there is a significant difference in the RTT obtained from different RRC states of the same UE, resulting in a non-zero RTT difference. For example, when two packets P_1 , P_2 arrive sequentially at the same UE in different RRC states, assuming P_1 with IDLE and P_2 with CR, the RTT difference between P_1 and P_2 will be $T_{IDLE} - T_{CR} = T_2$, which is non-zero. The RTT difference varies with the 4 RRC states (CR, ShortDRX, LongDRX, IDLE), so there are 16 RTT differences in LTE.

However, in fixed networks without RRC (e.g., DSL, fiber or Wi-Fi), data is transmitted directly to the target with little waiting. There is no significant difference in RTT between packets, and the expected value of the RTT difference is zero. Therefore, it is possible to use the characterization of RTT difference between cellular and fixed-line devices to identify devices in mobile networks from those in fixed networks.

Although we only cover the RRC mechanism for LTE/4G in this section, almost all modern mobile communication systems contain a similar RRC mechanism. The approach discussed in this paper also applies to 2G, 3G, 5G, Narrowband IoT, etc.

IV. LATENCY-BASE CELLULAR SUBNETS IDENTIFICATION

Based on the characteristics of RTT difference for downlink data on mobile networks, this section describes our latency-based approach to identify whether IPv6 /48 subnets are accessing links over cellular or fixed lines. First, we describe the method for triggering the RRC state transition and obtaining non-zero RTT differences with packets. Second, we show that the RTT difference in the /48 subnets is sufficient to identify cellular subnets. Finally, we present the features to characterize the RTT difference between cellular and fixed /48 subnets.

A. Triggering Non-zero RTT Difference with Packets

For description, we describe some of the symbols used. Suppose we send $n + 1$ probe packets ($P_0, P_1, \dots, P_n$) to a target and record the corresponding RTTs, $T_{P_0}, \dots, T_{P_n}$. The $T_{interval}$ represents the sending interval between packets. We use $state_i$ to represent the RRC state before packet P_i triggers the transition (the moment P_i arrives at PGW in Fig.1). When P_i reaches the UE, the UE state is always CR. Then before P_{i+1} arrives PGW, the state will likely transition to another

state $state_{i+1}$. We denote the state transition from $state_i$ to $state_{i+1}$ as:

$$state_i \xrightarrow{t_i} CR \xrightarrow{t'_i} state_{i+1}, i \in \{0, \dots, n\} \quad (2)$$

where t_i represents the cache latency introduced by state transition, t'_i represents the duration between CR and the moment P_{i+1} arrives at PGW ($state_{i+1}$). According to (2), $state_i$ determines t_i and t'_i determines $state_{i+1}$, which we represent as $state_i \sim t_i$ and $t'_i \sim state_{i+1}$. We use $T'_{interval}$ to represent the interval between the $state_i$ and $state_{i+1}$ and $T'_{interval} = t_i + t'_i$.

The purpose of triggering RRC transition with packets is to make $state_i$ and $state_{i+1}$ different, which will cause T_{P_i} and $T_{P_{i+1}}$ to be different according to (1), resulting in a non-zero RTT difference.

1) *The sending interval $T_{interval}$ matters:* First, we show that $T'_{interval}$ is approximately equal to $T_{interval}$. $T_{interval}$ represents the time between the moments when P_i and P_{i+1} are sent from point M, and $T'_{interval}$ represents the time between the moments when P_i and P_{i+1} arrive at PGW. And the difference between $T'_{interval}$ and $T_{interval}$ is the time it takes for the packet to go from M to PGW, which is the T_0 in Fig.1. The T_0 is the propagation latency of the packet from point M to PGW, which is similar for P_i and P_{i+1} due to the stable propagation latency of the backbone in a short period (e.g., one second). So, we have

$$T_{interval} \approx T'_{interval} = t_i + t'_i \quad (3)$$

Thus, we can control the $T_{interval}$ to maintain the t_i and t'_i and make the RTT different. According to (1)(2)(3), we can send packets at the same $T_{interval}$ but different $state_i (\sim t_i)$ or different $T_{interval}$ but same $state_i (\sim t_i)$ to make t'_i different, which determines $state_{i+1} (\sim T_{P_{i+1}})$, making the T_{P_i} and $T_{P_{i+1}}$ different.

2) *Triggering RRC transition with packets:* As the sending interval makes the RTT different, we discuss how to send packets to achieve different RTTs below.

Packet P_0 . The main goal of P_0 is to discover the IP address of the IPv6 periphery described in Section II. The P_0 also triggers the RRC state transition ($state_0 \rightarrow CR$). The RTT of P_0 does not accurately reflect the latency introduced by the state transition, as routing table lookup or loopback in periphery routers [7] introduces additional delays. Therefore, we drop the RTT of P_0 . In the implementation, we do not limit the moment we send P_0 since the primary goal of P_0 is to discover the IPv6 address of peripheries.

Packet P_1 . We send P_1 to the addresses that P_0 discovered to trigger the transition from $state_1$ to CR . We always send P_1 at the same moment, which makes the $T_{interval}$ between P_0 and P_1 different for different targets and results in different $state_1$, shown in Fig.3. Since $state_1$ may be in one of the four RRC states, we can theoretically measure four possible T_{P_1} .

Packet P_i . The $P_i (i \in \{2, \dots, n\})$ has the same goal as P_1 but differs in $T_{interval}$, which is variable between P_0 and P_1 but fixed between $P_1, P_2, \dots, P_n$, shown in Fig.3. The purpose

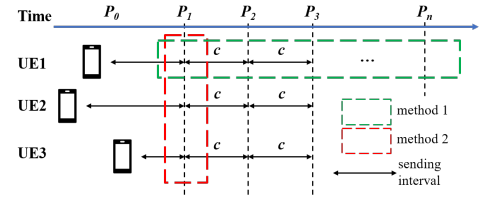


Fig. 3. We measure RTT by sending $n+1$ packets to each UE with the sending interval $T_{interval}$. There are two methods to measure the RTT difference between packets, each with different measurement overheads.

of $P_i (i \in \{2, \dots, n\})$ with the same $T_{interval}$ triggers the same state transition, so the RTT difference will be non-zero.

So, with the different $T_{interval}$ between P_0 and P_1 and the same $T_{interval} = c$ between P_i and P_{i+1} ($i \in \{1, \dots, n\}$), the RTT between packets will be different, making RTT difference non-zero.

3) *Measuring RTT difference:* There are two methods to measure the RTT difference between packets. The first one is the most prevalent, which measures the difference between packets to the same UE (the method1, green dashed box in Fig.3). Theoretically, we only need to send two packets to the same target to measure the RTT difference, such as P_1 and P_2 . But more packets are required due to the packet loss. In [10], [11], ten packets are needed, which is inefficient and may cause damage to target devices [13], [17]. The Zero RTT differences between UEs will likely affect the accuracy of Method 1.

The second method, which we proposed, measures the difference in RTT between packets to different UEs (the method2, red dashed box in Fig.3). The problem is that if the network paths of separate IPv6 addresses are significant different, the RTT difference will be meaningless due to the impact of varying network path lengths. We solve this problem by measuring the RTT difference in the /48 subnet. The /48 subnets are always assigned to the same end site [18], region access network [8], or subnets [3]. So the network paths of packets send to IPs in the same /48 subnet will be similar and possibly even the same.

Measuring RTTs in the /48 subnet benefits a lot. Firstly, the RTT difference across IPs in the /48 subnet eliminates the impact of network paths. Second, we can measure the RTT difference for each IPv6 by sending only one packet (P_1 to measure the RTT), which is 1/10 of the measurement overhead compared to [10], [11]. Finally, when combined with IPs in the /48 subnet, the loss of several packets does not affect the calculation of the RTT difference, which is robust to RTT loss.

B. The RTT Difference in /48 Subnets

To show the RTT difference of cellular /48 subnets in mobile networks is fundamentally different from fixed /48 subnets, we measure the RTT difference of the /48 subnets in different operators in China. We show two fixed and three mobile /48 subnets of them in Fig.4.

Fig.4 shows the CDFs of the RTT difference between subnets in three different China operators. We find the fixed and cellular subnets have different characteristics in RTT difference, such

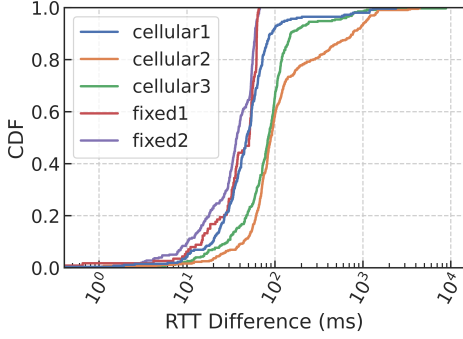


Fig. 4. The CDFs of the RTT difference between /48 subnets with different access types in three China operators. The fixed and cellular subnets have different characteristics. We used $\{T_i - \min T_i\}$ to calculate the RTT difference.

as the RTT difference range, variance, distribution in different time intervals, etc. Thus, we use the characteristics of RTT difference of /48 subnets to identify cellular subnets.

C. Features

Based on the observations, we describe the 20 features we used to identify cellular /48 subnets. Suppose N RTTs in a /48 subnet consist the $\mathbf{R} = \{T_{P_i}, i \in \{1, \dots, N\}\}$.

1) *The variance and range of the $\mathbf{R}$* : It yields two features;

2) *The distribution of RTT difference*: The RTT difference $\mathbf{D_R} = \{T_{P_i} - \min T_{P_i}\}$ eliminate the effort of network path. The variance, maximum, minimum, 95th and 5th percentile, and mean of $\mathbf{D_R}$ are calculated. It yields six features;

3) *The distribution of $\mathbf{D_R}$ between different percentiles*: The $\mathbf{D_R}$ of 100 percentile points are selected, $\{p_1, \dots, p_{100}\}$. We calculate the difference between these points, $\mathbf{S} = \{p_1 - p_2, \dots, p_{99} - p_{100}\}$. We calculate the variance, maximum, minimum, 95th, and 5th percentile, and mean of $\mathbf{S}$, which yields six features;

4) *The distribution of $\mathbf{D_R}$ in different intervals*: The $\mathbf{D_R}$ in the interval $[-t, t]$ is $\mathbf{G}_t = \{-t \leq \mathbf{D_R} \leq t\}$. We calculate $\mathbf{G} = \{\frac{|\mathbf{G}_t|}{N}\}, t \in \{10, \dots, \frac{g}{10}, \dots, g\}$. We set $g = 1,000$ ms, indicating we want to obtain the distribution in $[-1000$ ms, 1000 ms] in a step of 10 ms. We calculate the variance, maximum, minimum, 95th, and 5th percentile, and mean of $\mathbf{G}$. It yields six features.

V. EXPERIMENTAL EVALUATION

A. Dataset

Through a random probe of the known IPv6 prefixes, we collect the IPv6 /48 subnets containing at least one response IPv6 address (fixed or cellular). Then, we brutally scan the /48 subnets for all 65536 /64s to obtain active IP addresses. We measure the RTTs of the IPs and extract the features for each /48 subnet. Finally, we label the /48 subnets, forming the train and validation sets.

TABLE I
PREFIX ASSIGNMENT STRATEGY IN CHINA'S OPERATORS.

Operator	Access Type	/24 Prefix
China Telecom	fixed	240e:0300::/24
	cellular	240e:0400::/24
China Mobile	fixed	2409:8a00::/24
	cellular	2409:8900::/24
China Unicom	fixed	2408:8200::/24
	cellular	2408:8400::/24

1) *The method to measure RTTs in a /48 subnet*: We introduce the method of measuring the RTTs of the /48 subnet. Using ICMPv6 protocol, we measure the RTTs of a /48 subnet in two steps: First, brutally scanning the /48 subnet with 65536 /64 with the P_0 and discovering the active IPv6 addresses; Then for each IPv6 address discovered in the /48 subnet, we simultaneously send P_1 to measure the RTTs; The sending interval between P_0 and P_1 is at least 2 seconds. The UDP or TCP also works [13] depending on operators' firewall rules, which discussed in Section VII.

2) *Training set*: Via the open resources [19], [20] and the report [21], we summarise the IPv6 prefix assignment strategy for mobile and fixed broadband in China's major operators in Table I. The operators separate the different types of broadband networks into different /24 prefixes for ease of management. We verify the assignment strategy through lots of forum posts and with our devices.

By randomly probing 100,000 /64s (0.0233%) within each /32 prefix in Table I, we obtain 2,730 /48 subnets with at least one active IPv6 address, forming the candidate subnets. We label the /48 subnets as cellular if it is in the correspondent /24 mobile prefix of Table I. We label fixed subnets in the same way. Then we measure the RTTs in the candidate subnets and extract the features for each /48 subnets. Finally, we get 603 cellular and 689 fixed /48 subnets, forming the training set.

3) *Validation set*: RTR-NetTest [23] is a website developed by the Austrian telecoms regulator, the Regulatory Authority for Broadcasting and Telecommunications (RTR). It provides users with a quality of service measurement service and makes historical measurement data publicly available. We collect data from January 2022 to October 2022, extracting connection type (WLAN, 4G, 5G, etc.), address translation type (no_nat_ipv6), and anonymous IP (/48 subnet) to obtain 6978 cellular and 13048 fixed /48 subnets. The 6978 cellular subnets distribute in 150 ASes worldwide. Then we measure the RTTs of the /48 subnets. We label subnets as cellular for connection types with 4G, 5G, etc., and as fixed for connection types with WLAN, etc.

However, we find 1102 (5.5%) subnets are double-tagged (both tagged with fixed and cellular). Rula et al. [3] show that cellular tags contain very few false positives, but the fixed will be tagged incorrectly due to the prevalence of tethering. So, we label the "double-tagged" subnets as cellular. For the fixed subnets, We label them using three different ways and get different validation sets:

TABLE II
CLASSIFICATION ACCURACY OF OUR APPROACH.

dataset	TP	FP	TN	FN	Precision	Recall
Validation set 1	1847	128	461	7	93.52%	99.62%
Validation set 2	1847	1	878	7	99.95%	99.62%
Validation set 3	1847	1	258	7	99.95%	99.62%

- Validation set 1 (Fixed). We label the subnets as fixed when the subnet is fixed-tagged but not double-tagged.
- Validation set 2 (EUI64). The EUI64 are widely used in fixed-line devices [15], [22] but rarely used in cellular according to the measurement results of [7]. So, we calculate the percentage of EUI64 addresses in response addresses for each subnet and filter the subnets in which the EUI64 percentage is greater than 90% as fixed.
- Validation set 3 (Fixed&EUI64). We label the subnets as fixed when the subnet is fixed-tagged while the EUI64 percentage is greater than 90%.

B. Training

We used machine learning algorithms to learn the features of RTT difference in /48 subnets. We divide the training set into two sets in 7:3. We train the identification model on the first set and test it on the second set with different machine learning algorithms, such as Decision Tree, Random Forests, Support Vector Machines, and KNeighbors, etc. We find that the Random Forest performs the best, with a precision of 100% and recall of 100%. So, we select Random Forest to train the model finally. The results show the effectiveness of the identification model on /48 subnets in operators in China.

C. Validation

To validate the performance of the identification model we trained on subnets in China's operators, we use the validation sets to evaluate the model's performance on /48 subnets worldwide. We summarize the results in Table II, in which True positives (TP) represent correctly identified cellular subnets, and true negatives (TN) as correctly identified fixed subnets. False positives (FP) denote fixed subnets identified as cellular, and false negatives (FN) represent cellular subnets we inaccurately identified as fixed.

We show that our proposed method trained on subnets of China's operators is also effective on subnets worldwide. From Table II, we find that the Validation set 1, which tags the fixed subnets with original fixed tags, achieves a relatively low precision (93.52%). The reason is that although we filter out double-tagged subnets, some cellular subnets providing tethering are still incorrectly labeled as fixed but not double-tagged, making the FP high. To verify these, we manually check the 128 FP /48 subnets and find that 90% of them were "fixed" subnets surrounded by cellular subnets in a /44 or /42 subnet. We consider them false tags in RTR-NetTest that violate ease of management. While filtered with EUI64, the precision improves to 99.95% and the recall remains.

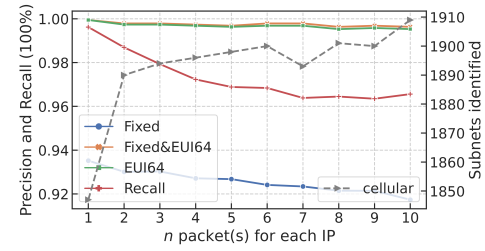


Fig. 5. The performance of the identification method with n probing packet(s) for each IP.

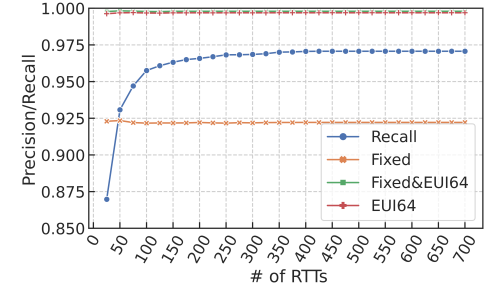


Fig. 6. The performance of the identification method with m RTTs in the /48 subnet.

D. Discussion

1) *How many packets for each IP are sufficient?*: We consider how many probe packets we need to send to each IP in the /48 subnet to perform well. We send ten packets with sending interval of 500 ms to each IP and measure the RTTs in each subnet in train and validation sets. The identification model is retrained and evaluated for each n ($n = 1, 2, \dots, 10$).

We show the results in Fig.5. When $n = 1$, the precision and recall are highest. As the n increases, the precision and recall decrease slightly while the number of identified cellular subnets increases by 3.24%. Considering the overhead of RTT measurements, we believe that $n = 1$ is the best choice. The additional probes per target do not bring new information.

2) *How many RTTs for each /48 subnet are sufficient?*: When packet loss happens, we discuss the number of RTTs a /48 subnet needed to identify the cellular subnet correctly. We simulate packet loss by controlling the number of measured RTTs in the /48 subnet. We randomly selected m ($m = 25, 50, \dots, 700$) RTTs in the /48 subnet in the validation sets and re-validated the identification model. For example, $m = 25$ implies that we only measure 25 RTTs under the /48 subnet, and consider the other RTTs lost.

When packet loss occurs, we show that the identification model can identify cellular /48 subnets with high precision and recall. Fig.6 shows the robustness of the identification model to packet loss. The precision of the identification model stays around 99.8%, which is almost unaffected by the packet loss; Despite packet loss in the /48 subnet, a few RTTs can achieve a high recall, e.g., 95.75% with 100 RTTs and 96.08% with 125 RTTs; When the number of RTTs reaches 400, the recall

comes to 97.06% and does not change anymore.

In addition, based on the robustness of packet loss, we can further reduce the measurement overhead by actively simulating packet loss. For example, 100 RTTs is sufficient to achieve 95.75% precision, so we do not need to measure the RTTs of all devices in the /48 subnet because not sending packets implies active packet loss.

VI. ETHICAL CONSIDERATIONS

The traffic to the UE in mobile networks is expensive, particularly the Cellular-IoT [17]. So probing with significant traffic may harm the target (e.g., over-billing [17]). To reduce the impact on cellular targets, we design a robust method to discover an IP address and measure its RTT with only two probe packets (64 bytes for each). Also, to minimize the impact on the network infrastructure, we use a sending rate of 6Mbps to send probe packets.

VII. LIMITATION

Firewall rules will likely drop the probe packets we send to /48 subnets [13]. When using ICMPv6 probes, only 26.47% of the cellular subnets in the RTR-NetTest have responses. We tested all cellular subnets using UDP, TCP, and ICMPv6, and most of the subnets that respond to ICMPv6 probes react to all three protocols. However, in our tests, T-Mobile (AS21928) dropped ICMPv6 probes but not UDP and TCP probes. Some cellular ASes don't respond to all our probes, e.g., AS1239 (T-Mobile US Sprint), and AS6805 (Telefonica Germany o2). We consider that the cellular ASes deploy a firewall to filter scanning probes initiated from the Internet.

VIII. CONCLUSION AND FUTURE WORK

This paper proposes an accurate latency-based method to identify IPv6 cellular /48 subnets. By evaluating a global /48 subnets dataset, our method achieves a precision of 93.52%~99.95% and a recall of 99.96% with at least 1/10 of the measurement overhead of existing methods and is robust to packet loss. In the future, we will use this method to launch a global measurement of cellular subnets identification to study the location of IPv6 cellular subnets on the Internet. The measurement results will help researchers understand different mobile operators' cellular subnet distribution characteristics, infer IPv6 subnet allocation strategies, etc.

ACKNOWLEDGMENT

This research is supported by the National Key Research and Development Program of China (2020YFE0200500) and the National Science Foundation of China (No. 62102020, No. 62002009).

REFERENCES

- [1] Jennifer Bly, "Mobile Edge of the Internet is Rapidly Moving to IPv6", 2020, <https://www.arin.net/blog/2020/01/16/mobile-edge-of-the-internet-is-rapidly-moving-to-ipv6/>
- [2] APNIC, "Deploy IPv6 - Success Stories", 2022, <https://www.apnic.net/community/ipv6/deploy-ipv6/#success>.
- [3] J. P. Rula, F. E. Bustamante, and M. Steiner, "Cell spotting: Studying the role of cellular networks in the internet," in Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC, 2017, pp. 191–204.
- [4] D. Plonka and A. Berger, "Temporal and spatial classification of active IPv6 addresses," in Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC, 2015, vol. 2015-October, pp. 509–522.
- [5] F. Li and D. Freeman, "Towards A User-Level Understanding of IPv6 Behavior," in Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC, 2020, pp. 428–442.
- [6] G. Song, J. Yang, L. He, Z. Wang, G. Li, C. Duan, Y. Liu and Z. Sun. "AddrMiner: A Comprehensive Global Active IPv6 Address Discovery System," in Proceedings of the USENIX Annual Technical Conference, ATC, 2022, pp. 309–326.
- [7] X. Li, B. Liu, X. Zheng, H. Duan, Q. Li, and Y. Huang, "Fast IPv6 Network Periphery Discovery and Security Implications," in Proceedings of the Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN, 2021, pp. 88–100.
- [8] Z. Zhang, A. Marder, R. Mok, B. Huffaker, M. Luckie, K. C. Claffy and A. Schulman, "Inferring regional access network topologies: Methods and applications," in Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC, 2021, pp. 720–738.
- [9] M. Ziv, L. Izhikevich, K. Ruth, K. Izhikevich, and Z. Durumeric, "ASdb: A system for classifying owners of autonomous systems," in Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC, 2021, pp. 703–719.
- [10] Y. Lee and N. Spring, "Identifying and analyzing broadband internet reverse DNS names," in the Proceedings of the International Conference on emerging Networking EXperiments and Technologies, CoNEXT, 2017, pp. 35–40.
- [11] R. Padmanabhan, P. Owen, A. Schulman, and N. Spring, "Timeouts: Beware surprisingly high delay," in Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC, 2015, pp. 303–316.
- [12] N. Larson, D. Baltrunas, A. Kvalbein, A. Dhamdhere, K. C. Claffy, and A. Elmokashfi, "Investigating excessive delays in mobile broadband networks," in Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges, 2015, pp. 51–56.
- [13] H. Hong et al., "When Cellular Networks Met IPv6: Security Problems of Middleboxes in IPv6 Cellular Networks," in Proceedings of the IEEE European Symposium on Security and Privacy, EuroS&P, 2017, pp. 595–609.
- [14] E. Rye, R. Beverly, and K. C. Claffy, "Follow the scent: Defeating IPv6 prefix rotation privacy," in Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC, 2021, pp. 739–752.
- [15] E. Rye, and R. Beverly, "IPv6SeeYou: Exploiting Leaked Identifiers in IPv6 for Street-Level Geolocation". arXiv preprint arXiv:2208.06767, 2022, n. pag.
- [16] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4G LTE networks," in Proceedings of the International Conference on Mobile Systems, Applications, and Services, MobiSys, 2012, pp. 225–238.
- [17] S. Wang, G. Tu, X. Lei, T. Xie, C. Li, P. Chou, F. Hsieh, Y. Hu, L. Xiao, and C. Peng, "Insecurity of operational cellular IoT service: new vulnerabilities, attacks, and countermeasures." in Proceedings of the Annual International Conference on Mobile Computing and Networking, 2021, pp. 437–450.
- [18] IAB and IESG, "IAB/IESG Recommendations on IPv6 Address Allocations to Sites", RFC 3177, September 2001, <https://www.rfc-editor.org/info/rfc3177>.
- [19] Z. Lin, "IPv6 address assignment table of the three major operators in China," 2022, <https://linzening.fe80.cn/pages/ipv6/i6.html>.
- [20] ZX Inc, "IPv6 address query tool," 2022, <http://ip.zxinc.org/ipquery/>.
- [21] J. Gao, S. Gao, X. Wu, "New Ways of IPV6 Scanning," 2021, <https://www.blackhat.com/eu-21/briefings/schedule/#new-ways-of-ipv-scanning-24928>.
- [22] E. C. Rye and R. Beverly, "Discovering the IPv6 network periphery," in Passive and Active Measurement, PAM, 2020, pp. 3–18.
- [23] RTR, "RTR - NetTest," 2022, <https://www.netztest.at/>.
- [24] 3GPP, "TS 36.331, Radio Resource Control (RRC); Protocol specification (Release 17)", 2022.
- [25] 3GPP, "TS 23.060, General Packet Radio Service (GPRS);Service description (Release 17)", 2022.