Modeling and Generating Control-Plane Traffic for Cellular Networks

Jiayi Meng
Purdue University
meng72@purdue.edu

Jingqi Huang
Purdue University
huan1504@purdue.edu

Y. Charlie Hu
Purdue University
ychu@purdue.edu

Yaron Koral
AT&T Labs
yk216h@att.com

Xiaojun Lin
Purdue University
linx@purdue.edu

Muhammad Shahbaz
Purdue University
mshahbaz@purdue.edu

Abhigyan Sharma
AT&T Labs
abhigyan.sharma@gmail.com

ABSTRACT

With 5G deployment gaining momentum, the control-plane traffic volume of cellular networks is escalating. Such rapid traffic growth motivates the need to study the mobile core network (MCN) control-plane design and performance optimization. Doing so requires realistic, large control-plane traffic traces in order to profile and debug the mobile network performance under real workload. However, large-scale control-plane traffic traces are not made available to the public by mobile operators due to business and privacy concerns. As such, it is critically important to develop accurate, scalable, versatile, and open-to-innovation control traffic generators, which in turn critically rely on an accurate traffic model for the control plane. Developing an accurate model of control-plane traffic faces several challenges: (1) how to capture the dependence among the control events generated by each User Equipment (UE), (2) how to model the inter-arrival time and sojourn time of control events of individual UEs, and (3) how to capture the diversity of control-plane traffic across UEs. We present a novel two-level hierarchical state-machine-based control-plane traffic model. We further show how our model can be easily adjusted from LTE to NextG networks (e.g., 5G) to support modeling future control-plane traffic. We experimentally validate that the proposed model can generate large realistic control-plane traffic traces. We have open-sourced our traffic generator to the public to foster MCN research.

CCS CONCEPTS

• **Networks** → **Mobile networks**; **Network performance modeling**; **Network measurement**; **Network simulations**; *Network management*; *Network monitoring*.

KEYWORDS

4G/5G; mobile core network; control plane; traffic modeling and synthesis

ACM Reference Format:
Jiayi Meng, Jingqi Huang, Y. Charlie Hu, Yaron Koral, Xiaojun Lin, Muhammad Shahbaz, and Abhigyan Sharma. 2023. Modeling and Generating Control-Plane Traffic for Cellular Networks. In *Proceedings of the 2023 ACM Internet Measurement Conference (IMC '23), October 24–26, 2023, Montreal, QC, Canada.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3618257.3624808

1 INTRODUCTION

Motivation. The Mobile Core Network (MCN) is at the heart of the cellular network; it manages and tracks all users' activity (including mobility) as well as forwards data traffic between users and the Internet. To efficiently and flexibly handle control and data traffic in these networks, Control-/User-Plane Separation (CUPS) was introduced in 3GPP Release 14 for 4G [1] and refined in 3GPP Release 15 for 5G [4], to separate cellular operations between a control plane and a data plane to process control and data traffic, respectively.

In the past few years, as 5G deployment has been gaining momentum, cellular networks have witnessed not only an explosive growth in data-plane traffic, but also a significant increase in control-plane traffic. For example, the control-plane traffic has been reported to grow 50% faster than data-plane traffic since 2015 [57], and some carriers are reportedly experiencing more than 100× increase in the volume of transactions in the 5G control plane compared to 4G in 2021 [9]. Such control-plane traffic growth challenges mobile network operators and designers to innovate on mobile network architectural design not only for the data plane but also for the control plane to ensure high-quality mobile user experience.

However, the large body of recent works on LTE and 5G has focused on the data plane of MCN, i.e., on modeling data-plane traffic (e.g., [18, 24, 37, 44, 45, 48]) and improving data-plane performance and scalability [14, 38, 39, 52, 56, 58, 63]. Several recent works have studied the control plane of MCN, but has focused on limited control-plane event types, ignoring that real-world traffic comprises diverse event types and has intricate event dependence for each UE specified by 3GPP [13, 38, 39, 56, 63]. Therefore, it remains unclear how well their designs will perform with realistic large-scale control-plane traffic today as well as tomorrow for NextG.

We argue that for MCN to sustain the rapid growth in traffic demand in the coming years, it is equally vital and urgent to study the impact of control-plane traffic on the MCN performance and consequent design improvement. And, conducting such studies

IMC '23, October 24–26, 2023, Montreal, QC, Canada

ACM ISBN 979-8-4007-0382-9/23/10...$15.00
https://doi.org/10.1145/3618257.3624808

critically relies on high-fidelity large-scale control-plane traffic to drive the MCN in order to evaluate and validate MCN designs under realistic workloads.

Despite the need for large-scale control-plane traffic traces in MCN, such data is only accessible by mobile network operators, who are reluctant to directly share their traffic traces due to business and privacy concerns. As a result, the lack of public MCN control-plane traffic hinders the in-depth study of MCN design and performance optimization by the broad networking and systems communities. While previous work [24] studied the control-plane traffic, it has mostly focused on theoretical traffic modeling without looking into real-world traffic behavior.

Our contributions. This paper makes four contributions toward modeling and developing a readily-usable generator for the control-plane traffic of LTE/5G mobile networks.

(1) Understanding inapplicability of traditional traffic models. We first study whether traditional probability distributions, which have been widely used for modeling Internet traffic, can be readily used to model the control traffic originated by individual UEs in the mobile network. We randomly sample 37, 325 real UEs (consisting of three primary types of devices, i.e., phones, connected cars, and tablets) from one major carrier in the US. We collect their control-plane events under LTE over one whole week (196, 827, 464 events in total). We choose to study LTE, as 5G deployment is still at an initial stage and systematic extensive trace collection has not been done by mobile operators. We then perform two standard statistical tests, the Kolmogorov-Smirnov test [54] and the Anderson-Darling test [67]. Our statistical test results show that surprisingly, the inter-arrival time of the control events and the duration of UEs continuously staying in each one of the four UE states (i.e., sojourn time) of EPS[1] Mobility Management (EMM) and EPS Connection Management (ECM) for the mobile network cannot be modeled as Poisson processes or other traditional probability distributions, including Pareto [34], Weibull [66], and Tcplib [25, 26].

We further study the reasons why these traditional probability distributions fail to model control traffic by analyzing how well the Poisson distribution can model the burstiness of control-plane traffic via variance-time plots [31, 43], and directly comparing the cumulative distributions of the trace with the fitted Poisson distributions. Our analysis reveals that the control-plane traffic of the mobile network has much higher burstiness and longer tails in their cumulative distributions, compared to the traditional probability models.

(2) Modelling control-plane traffic, the right way. Designing an accurate, scalable, versatile, and open-to-innovation control-plane traffic model for cellular networks poses several key challenges: (1) Different types of control-plane events of a UE have intricate dependence on each other, which cannot be easily captured by traditional models. (2) Our measurement study above shows that traditional probability distributions also fail to effectively model the inter-arrival time of the events and the sojourn time of the UE states of EMM and ECM for individual UEs. (3) Our measurement study also shows that control-plane events exhibit significant diversity in device types, the time-of-the-day, and across different UEs; a single model is unlikely to capture such diversities appropriately.

In this paper, we propose a two-level hierarchical state-machine-based traffic model, which addresses the above modeling challenges with three key components. (1) We extend the EMM and ECM state machines specified by 3GPP into a two-level state machine to capture the dependence among control events. (2) We leverage the Semi-Markov Model to model the sojourn time and the transition probability for the states in the two-level state machine, which overcomes the limitation of the Markov model by assuming that the sojourn time in the same state follows the exponential distribution with a constant hazard rate. (3) To capture the traffic diversity in device types, the time-of-the-day, and across UEs, we develop an adaptive clustering scheme to effectively cluster the UEs based on their traffic similarity for each combination of (hour, device-type) tuple. Finally, we instantiate the parameters of our model using a sample traffic trace for every combination of (UE-cluster, hour, device-type).

(3) Extending the control-plane traffic model for NextG networks. We present a methodology to showcase how to easily adjust our proposed traffic model for LTE to NextG (i.e., 5G); the parameters of the new traffic model for 5G can be readily seeded with a large-scale control-plane trace for 5G when it is available, or directly scaled from the 4G model.

(4) Experimental validation of proposed control-plane traffic model. We develop a baseline and two variations of our method using the fitted Poisson distributions without or with leveraging our adaptive clustering scheme or two-level state machine for LTE. We compare the synthesized traces using our method and the other three methods against two real traces for 38K and 380K UEs, respectively.

We show that our method outperforms the other three methods from both macroscopic and microscopic perspectives: (1) We first compare the breakdown of the synthesized events with that of the real events. Compared with the real traces, our synthesized traces achieve small differences, i.e., within 1.7%, 5.0% and 0.8%, for phones, connected cars, and tablets, respectively, while the traces generated by the baseline has the differences up to 47.8%, 47.7%, and 47.5%, for both UE population sizes. (2) We then compare the per-UE traffic behavior, including the numbers of events per UE and the sojourn time in the UE states for the two dominant state transitions (i.e., between `CONNECTED` and `IDLE`). Compared with the other three methods, for phones, our method reduces the maximum y-distance of the CDF of events per UE between the synthesized and actual traces by over 7.74×/7.46× for `SRV_REQ`/`S1_CONN_REL` events, and the maximum y-distance of the CDF of the sojourn time in `CONNECTED`/`IDLE` states between the synthesized and actual traces by over 4.77×/3.25×. Similar improvements are observed for connected cars, by 1.15×~2.65×, and for tablets, by 2.80×~8.56×.

We have open-sourced the developed control-plane traffic generator to the community to stimulate further research on MCN design and optimization for 4G/5G and beyond.[2]

[1] EPS stands for Evolved Packet System in LTE.

[2] https://gitlab.com/serverless-5g/cellular-network-control-plane-traffgen

Table 1: Breakdown of control-plane events of LTE for different types of devices in a 7-day trace (P: phones; CC: connected cars; T: tablets).

Event	Event Type	P	CC	T
Attach [2]	ATCH	0.1%	0.9%	1.2%
Detach [2]	DTCH	0.2%	0.9%	1.1%
Service Req. [2]	SRV_REQ	45.5%	38.9%	43.9%
S1 Conn. Rel. [2]	S1_CONN_REL	47.5%	45.2%	47.7%
Handover [2]	HO	3.8%	6.6%	2.1%
Tracking Area U. [2]	TAU	2.9%	7.4%	4.0%

2 BACKGROUND

We begin with a brief background of LTE network architecture and its control-plane events.

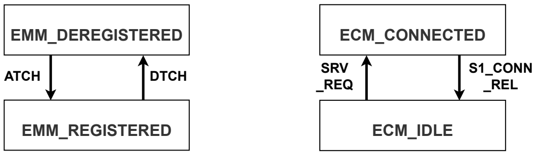
2.1 LTE Network Architecture

The LTE mobile network consists of three components: UE, Radio Access Network (RAN) and Evolved Packet Core (EPC), which finally connects to the Internet to provide data services to UEs. A UE is a device used by an end-user to communicate with the mobile network (e.g., phone, tablet, IoT, etc.). RAN resides between UEs and EPC and manages the radio spectrum of a distributed collection of base stations that directly communicate with UEs.

EPC represents the MCN of LTE. It consists of five network functions including Mobility Management Entity (MME), Home Subscriber Server (HSS), Policy and Charging Rules Function (PCRF), Packet Data Network Gateway (PGW), and Serving Gateway (SGW). To efficiently and flexibly handle control and data traffic, EPC is partitioned into a control plane and a user plane [2]. The control plane manages signaling traffic between RAN and MME and the other network functions of MCN (i.e., HSS, PCRF, SGW, and PGW). The user plane (also called the data plane) forwards data traffic among RAN, SGW, and PGW. Among the five network functions in MCN, MME is the main signaling function, which directly connects to UE/RAN in the control plane of LTE.

2.2 LTE Control-Plane Events

Table 1 summarizes major LTE control-plane events among UE, RAN, and MCN.[3] In the control plane [2], (1) *Attach* (ATCH) registers the UE with the MCN; (2) *Detach* (DTCH) deregisters the UE from the MCN, when the UE is switched off; (3) *Service Request* (SRV_REQ) creates a signaling connection for the UE to send/receive signaling messages or data; (4) *S1 Connection Release* (S1_CONN_REL) releases the signaling connection in the control plane, and other resources associated with the UE in the data plane; (5) *Handover* (HO) switches the UE from current serving cell to another cell; (6) *Tracking Area Update* (TAU) updates UE's tracking area, when the UE moves to another tracking area (comprising a new set of cells) or the periodic timer of TAU is expired or some other cases, e.g., the UE reselects to LTE from 3G or re-registers to LTE after the fallback for circuit-switched services, etc.



(a) EMM. (b) ECM.

Figure 1: UE state machines of LTE.

Dependence among events. The control events listed in Table 1 for a UE are not independent, as required to conform to the 3GPP protocol, regardless of whether the events are triggered by the UE activities, power outages of base stations, etc. The protocol specifies that one UE follows two independent state machines when interacting with MCN: EMM and ECM [2]. Some control-plane events, such as ATCH, DTCH, SRV_REQ, and S1_CONN_REL, can trigger changes to the UE states, while others cannot change the UE states but still have intricate dependence on those states and thus the corresponding events.

(1) The *EMM state machine* describes the UE's Mobility Management states that maintain the information related to UE's registration with the MCN. Figure 1a shows the two primary EMM states, EMM_DEREGISTERED and EMM_REGISTERED, which are denoted as DEREGISTERED and REGISTERED, and the corresponding control events that trigger transitions between them. Generally, the UE can be powered on (off), which in turn triggers ATCH (DTCH) event to enter REGISTERED (DEREGISTERED) state.
(2) The *ECM state machine* describes the signaling connectivity between the UE and the MCN, when the UE stays in REGISTERED. Figure 1b shows the two primary ECM states, ECM_CONNECTED and ECM_IDLE (denoted as CONNECTED and IDLE), and the corresponding control events that trigger state transitions. Generally, SRV_REQ (S1_CONN_REL) can be triggered to switch the UE state to CONNECTED (IDLE). For the other control-plane events, some (e.g., TAU) can happen in both CONNECTED and IDLE, while the rest (e.g., HO) can only happen in CONNECTED.

3 MOTIVATION AND DESIGN GOALS

We discuss the motivation and design goals for modeling and generating control traffic for cellular networks, and the rationale for modeling *individual* UE's control traffic.

3.1 Usage of Control-Plane Traffic Models

In addition to the motivations for developing control-plane traffic models discussed in §1, i.e., enabling evaluation of MCN architecture design and optimization, we elaborate on several other use cases of control-plane traffic models.

(1) **Monitoring MCN.** Network management critically relies on real-time monitoring of network traffic. It has been intensively studied recently for data flows (e.g., five-tuples), via either sampling-based [22, 60] or sketch-based telemetry [7, 17, 23, 35, 49, 50, 79, 81]. As control-plane traffic is gaining prominence in cellular networks, they also need to be monitored in real time. However, it is unclear how well the above schemes will perform for control-plane traffic in cellular networks. Accurate control-plane traffic models can help to develop effective monitoring

[3] We ignore events that happen only between UE and RAN.

schemes, e.g., with better accuracy and smaller memory footprints. For example, such models can help to determine a good sampling rate for sampling-based monitoring when collecting telemetry metrics.

(2) **Conducting large-scale simulations for NextG cellular networks.** Industry trend analysis [29] has projected that the number of devices (especially IoT devices) connected to the cellular network will grow significantly in the next 5 years. To evaluate the scalability of MCN design especially under the traffic load in the (near) future, we need control-plane traffic models that can capture the scaling of realistic traffic behavior (e.g., how they will grow in ensuing years). Those models enable the trace-driven analysis for large-scale simulations. The developed traffic models in this paper are already being actively used by the Aether community to study the scalability of Aether 5G core design [10].

3.2 Design Goals

To support the above usage, the mobile network's control traffic generator must meet the following requirements:

(1) **Accuracy:** The generator outputs realistic control-plane traffic for a fixed UE population, i.e., it can capture the inter-arrival duration of each type of events.
(2) **Event-Owner Labeling:** Every event in the generated control traffic needs to be labeled with its originating UE. This is required to properly drive the network functions of the MCN, e.g., a UE transition among EMM and ECM states in both 4G and 5G.
(3) **Scalability:** The generator should output realistic control traffic for an arbitrary UE population, e.g., for evaluating the scalability of an MCN design under increasing workload.
(4) **NextG Network Support:** The generator should support next-generation cellular networks, e.g., generating realistic control events from 5G UEs.

As with modeling Internet traffic, modeling control-plane traffic of cellular networks also needs to model the inter-arrival time of packets (i.e., control events). Unlike modeling Internet traffic, modeling control-plane traffic does not need to model the sizes of the control events, because each event has a fixed and small size, following the 3GPP specification.

3.2.1 Why not model aggregate control-plane traffic?

We envision the primary use of the control traffic generator is for monitoring or evaluating the performance of an MCN design, by driving the MCN with the aggregate control traffic due to a given UE population. As such, we could try to directly model the aggregate control traffic for a given UE population (e.g., the aggregate SRV_REQ events due to a set of UEs) by fitting the aggregate traffic for each event type in our trace collection using some well-known probabilistic distribution, similar to the prior work for modeling Internet traffic, and then use the fitted distribution to generate synthesized aggregate traffic of that event type for a given UE population.

However, modeling aggregate control-plane traffic across UEs has three limitations: (1) It is oblivious to and cannot capture the dependence among different control-plane events of individual UEs, e.g., that a SRV_REQ should happen after a S1_CONN_REL for a given UE. (2) Since the model only captures the inter-arrival time of the control events of each type for the set of UEs as whole, it will not be able to label each individual control event generated with a proper UE id, which is needed to correctly drive event processing performed by MCN functions, which is UE-oriented. (3) Since such a model is derived by fitting the control traffic trace for a fixed UE population (in the sample trace), it is often difficult to generate traffic for different UE population sizes, e.g., if the fitted model is a Pareto/Tcplib distribution.

For these reasons, we focus on modeling the control traffic of individual UEs, which will not have any of the above limitations and can be used to build a traffic generator that meets all the design requirements discussed in §3.2.

4 FAILURE OF CLASSIC PROBABILITY DISTRIBUTION-BASED MODELING

In this section, we study whether traditional probability distributions, which are widely adopted for Internet traffic, can model the control-plane traffic of individual UEs in cellular networks, and if not, why those distributions fail.

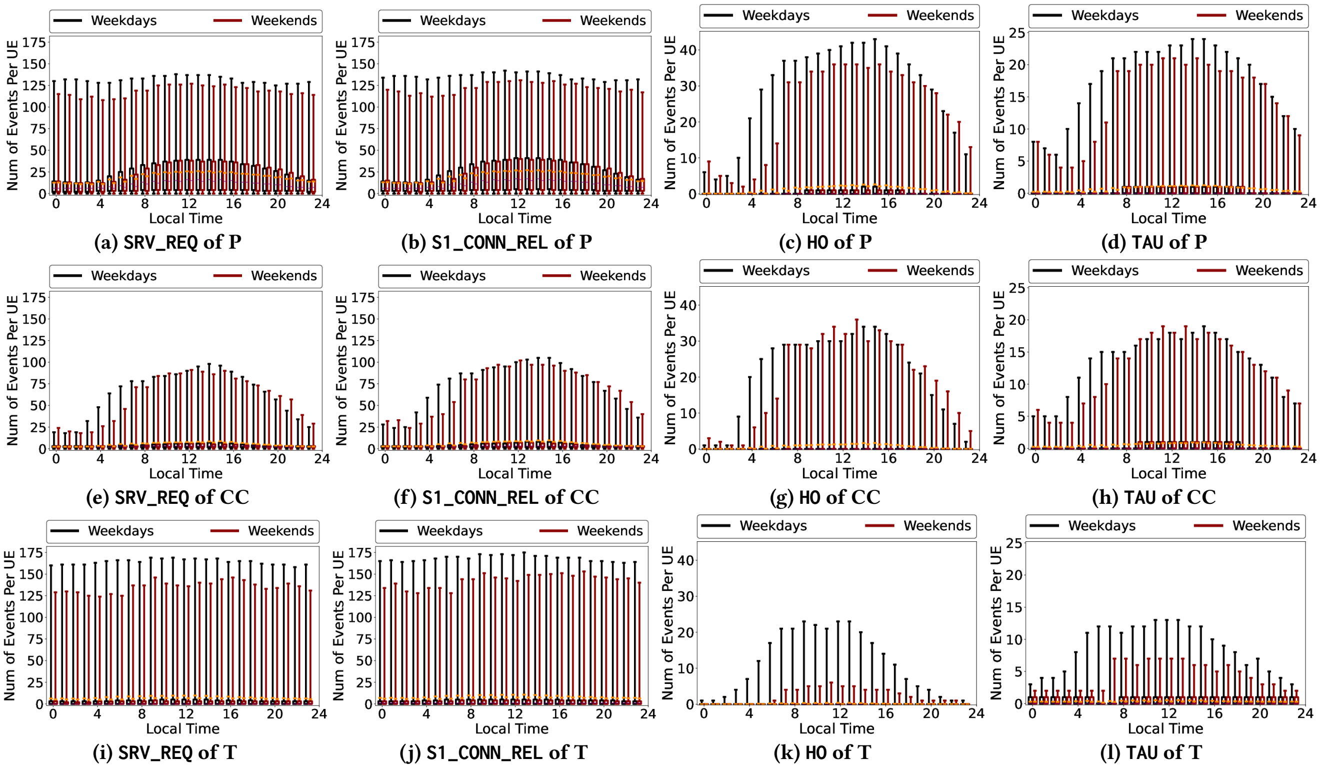
Dataset. Considering 5G deployment is still at a rudimentary stage and systematically collecting control-plane traces of MCN is not yet available, we choose to study an LTE trace. Specifically, we randomly sampled 37,325 real UEs from a major mobile carrier over the entire US and collected their control events recorded from MMEs over one whole week of June in 2022. The timestamps collected have a millisecond granularity. In total, we collected 196,827,464 events.

We also categorize the UEs into three primary types of devices: *phones*, *connected cars*, and *tablets*. We derive the device type of every UE via the Type Allocation Code (TAC), which is the first eight digits of UE's IMEI, and can identify the corresponding manufacturer and device model and thus the device type [77]. Of all sampled UEs, 23,388 are phones, 9,308 are connected cars, and 4,629 are tablets.

4.1 Can Classic Probability Distributions Model Individual UE Traffic?

Network traffic modeling has been a foundation area of research over the history of the Internet. Many probability distributions have been leveraged to model the inter-arrival time of Internet traffic, as summarized below.

(1) The *Poisson distribution* is the predominant model used for modeling network traffic arrivals [15, 30, 40, 59]. It has been proven valid for modeling the arrival time of user-initiated sessions in wide-area networks, e.g., TELNET connections and FTP control connections [59]. Specifically, a Poisson process characterizes the inter-arrival duration D_n as independently and exponentially distributed with a fixed rate parameter λ, i.e., $P(D_n > t) = e^{-\lambda t}$.
(2) The *Pareto distribution* [34] has been applied to model self-similarity in packet traffic of the wide-area network [6, 30]. It models the inter-arrival time by a power-law probability distribution that follows the probability density function of $f(x) = \alpha\, x_m^{\alpha}\, x^{-(\alpha+1)}$, where α is the shape parameter and x_m is the minimum possible value of x (normally, $x_m = 1$) [34].



(a) SRV_REQ of P

(b) S1_CONN_REL of P

(c) HO of P

(d) TAU of P

(e) SRV_REQ of CC

(f) S1_CONN_REL of CC

(g) HO of CC

(h) TAU of CC

(i) SRV_REQ of T

(j) S1_CONN_REL of T

(k) HO of T

(l) TAU of T

Figure 2: Box plots of numbers of control events per device-hour of different types of devices over 24 hours.

(3) The *Weibull distribution* [66] has been shown to capture the inter-arrival time dynamics from different Internet traffic levels (sessions, flows, and packets) [11]. A Weibull process follows the probability density function of $f(x) = \frac{k}{\lambda} (\frac{x}{\lambda})^{k-1} e^{-(x/\lambda)^k}$ for $x \geq 0$, where k is the shape parameter and λ is the scale parameter.
(4) The *TCPlib distribution* is an empirical distribution proposed to model the inter-arrival time within TELNET connections of real wide-area TCP/IP traffic [30].

However, whether those theoretical distributions can model LTE's control-plane traffic of individual UEs is unclear. We next answer this concern by addressing two following questions: (1) How to apply traditional probability distributions to the control-plane traffic? (2) How to validate if the traffic can be modeled using those distributions?

4.1.1 UE clustering to facilitate applying traditional probability distributions.

To properly model individual UE traffic using classical probability distributions, we first characterize the diversity of the control-plane traffic of LTE.

Diversity in device types, the time-of-day, and across UEs. To understand how the control-plane traffic is affected by the time-of-day among UEs, we divide the trace into 1-hour intervals for each day. We analyze the diversity of numbers of events per UE across different hours of the day. Specifically, within each 1-hour interval, we count number of events for every type of control events for each UE. For each device type and each event type, we draw the box plot for number of events per UE for all the UEs across different hours of the day. Each box plot comprises 24 boxes, where each box is defined by the lower and upper quartiles of the data, the whiskers at the two ends represent its minimum and maximum, the red line in the center of the box is the median and the orange line is the average.

Figure 2 shows that the per-UE traffic varies over different hour-of-day for different types of devices. Specifically, the average per device-hour volume of the four dominant event types (i.e., SRV_REQ, S1_CONN_REL, HO, and TAU) drops significantly from the peak hour to the slowest hour of the day by 2.27×∼86.15× for phones, by 3.43×∼1309.33× for connected cars, and by 1.45×∼90.06× for tablets.

Figure 2 also shows that among UEs of the same device type in the same hour, the differences between maximum and minimum numbers of those four dominant event types are also large across different hours of the day, i.e., 2∼142 for phones, 1∼105 for connected cars, and 0∼175 for tablets.

UE clustering to facilitate traffic modeling. The above control-plane diversity has two immediate implications. First, the diversity in device types and the time-of-the-day suggests a single probability distribution is unlikely to model the per-UE traffic across device types and times of the day. Thus we first preprocess the input control traffic trace by dividing the entire trace into non-overlapping 1-hour intervals for each device type. However, we find it is impractical to model the inter-arrival time for every type of control events for every single UE for each 1-hour interval, since each UE has a very limited number of events per hour.

Second, to overcome this too-few-events problem, we observe the events of different UEs are i.i.d. and hence we could merge the inter-event arrival time of all the UEs of the same device type for each hour and try to fit a probability distribution.[4] However, the traffic diversity across the UEs (of the same device type) suggests that trying to fit the merged control-plane traffic across all UEs of the same device type for the same hour with a single probability distribution is unlikely to work well.

Instead, we cluster the UEs of the same device type using the adaptive clustering scheme used in our proposed traffic model (discussed later in §5.3), so that the UEs in the same cluster will both have similar traffic characteristics and have enough data samples to facilitate fitting by a single well-known model. Since there also exist repetitive diurnal patterns of control traffic observed on both weekdays and weekends (Fig. 2), we also group the inter-arrival time of the same hour from different days together for each (UE-cluster, hour, device-type, event-type) combination. Finally, we try to fit the traffic in each combination with traditional probability models using Maximum Likelihood Estimation (MLE).

In addition to the six types of control events, we also consider the four UE states (i.e., `REGISTERED`, `DEREGISTERED`, `CONNECTED`, and `IDLE`), which cannot be captured by modeling each event type separately. We model the sojourn time in those states for individual UEs using traditional probability distributions. Specifically, for each UE, within each interval, we replay the traffic trace while following the EMM and ECM state machines specified by 3GPP (Fig. 1a and Fig. 1b), to calculate the duration of staying in each of the four UE states. For each (UE-cluster, hour, device-type, state) combination, we group the sojourn time over different UEs and days, and use MLE to fit the grouped sojourn time.

4.1.2 Validating traditional distribution-based modeling.

We first examine whether Poisson processes can model the inter-arrival time or the sojourn time for individual UE traffic in *each 1-hour interval per device type per UE-cluster*. Specifically, we apply two standard statistical tests, the Kolmogorov-Smirnov (K–S) test [54] and the Anderson-Darling (A^2) test [67]. The K–S test compares the maximum distance between the empirical cumulative distribution function (CDF) of the sample data and the theoretical CDF of the reference distribution (e.g., exponential distribution) [16]. It outputs the *K–S test statistic* (i.e., the supremum of the set of the distances) along with the corresponding *p-value*. A *p-value* of 0.05 or lower is considered statistically significant between the empirical distribution of the sample data and the reference distribution [76]. The A^2 test is a modification of the K–S test and gives more weight to the tails of the observation data [69]. It outputs the *A^2 test statistic*, the *critical values*, and the corresponding *significance levels* calculated for the reference distribution. The *A^2 test statistic* is used to compare against the *critical values* calculated specifically for the reference distribution to determine whether to accept or reject the null hypothesis under some *significance level*. In this paper, we focus on the *significant level* of 5%.

To examine fitting with other distributions (i.e., Pareto, Weibull and Tcplib), we repeat the same preprocessing procedure and only perform the K-S test to decide if the inter-event arrival time for each type of event per UE or the duration staying in each of the four UE states is drawn from one of those distributions. We skip the A^2 test, because it can only test against some common distributions at the moment (e.g., normal and exponential).

Results. Surprisingly, we find that although clustering UEs can increase the percentages of the 1-hour intervals that pass the K–S and A^2 tests for the exponential distribution from 0.0% to up to 23.8% for `ATCH` and `DTCH`, for the other event types and the four UE states below 3.0% of the 1-hour intervals pass the K–S and A^2 tests. This suggests that Poisson processes cannot model the inter-arrival time of all six types of events as well as the sojourn time in all four UE states for all three types of devices. We note that there are limit theorems in the literature [21] stating that the superposition of many independent point processes approaches a Poisson process. However, such results are for the inter-arrival time of the *aggregate* arrival process, and thus do not imply that the *individual* point process can be modeled as a Poisson process. Since, in this paper, we aim to model the inter-arrival time of individual UE (individual point process), our findings do not contradict with [21].

We also find that the inter-arrival and sojourn time cannot be modeled by other traditional probability distributions (i.e., Pareto, Weibull, and TCPlib). Additional details can be found in Appendix A.

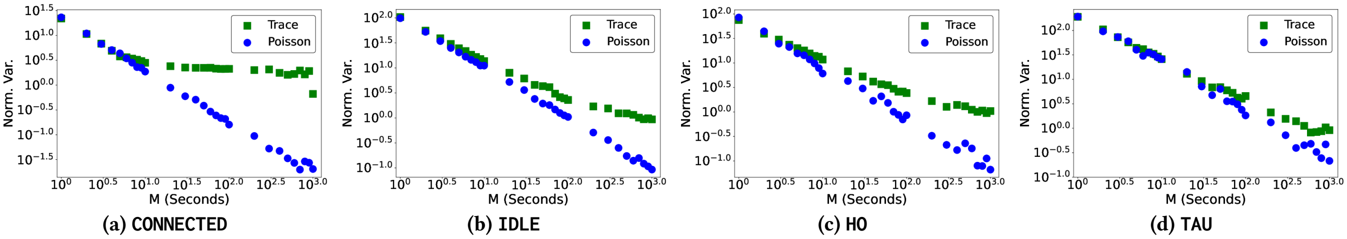
4.2 Why do Traditional Modeling Fail?

To understand why traditional probability distributions fail to model either the sojourn time in the four UE states or the inter-arrival time for different control events, we zoom into the two dominant UE states, `CONNECTED` and `IDLE`, and the two important types of control events, `HO` and `TAU`, and analyze: (1) How well the Poisson distribution can model an essential property of the control-plane traffic, its burstiness; (2) Whether the upper and lower tails of the observed inter-arrival time distribution can be captured by the Poisson distribution.

Burstiness. We first calculate the variance of numbers of events over different time scales, known as *variance-time plot* [31, 43], per hour, per device-type, to assess how well the Poisson distribution can model the burstiness of control-plane traffic. Specifically, we first divide the timeline into 100ms intervals. We count the number of events per 100ms interval for every type of device and event. Next, we consider different time scales M, ranging from 1 to 10^3 seconds. For each time scale of M seconds, we calculate the average number of events per 100ms for every M-second window i, denoted as k_i, and then calculate the mean and variance of this metric across all M-second windows, denoted as $\overline{k_i}$ and $\hat{k_i}$, respectively. We then normalize $\hat{k_i}$ by the square of $\overline{k_i}$ for the observation and the reference distribution, respectively.

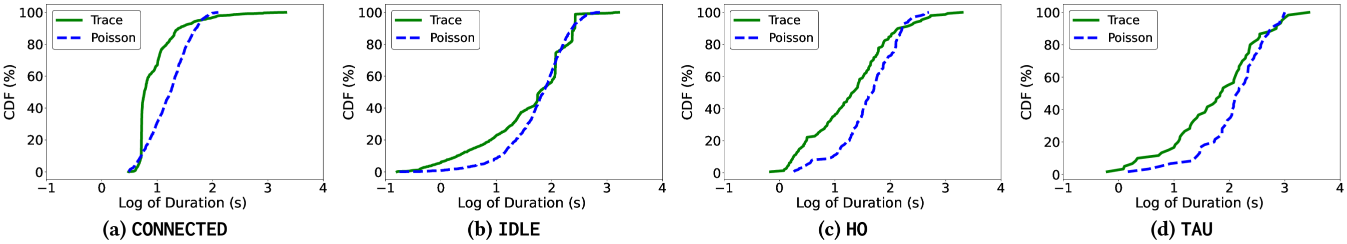
Figure 3 shows that for a randomly sampled cluster of phones, *the sojourn time in* `CONNECTED` *and* `IDLE` *exhibits stronger burstiness than the fitted Poisson models* across the time scale ranging from 10 to 10^3 seconds. The differences in the log-scale normalized variance between the real-world trace and the fitted exponential distribution are 0.43~2.00 and 0.18~1.00 for `CONNECTED` and `IDLE`, respectively. For `HO` and `TAU`, although they happen much less frequently than `SRV_REQ` and `S1_CONN_REL`, which trigger the state

[4] We note that merging the inter-arrival time of UEs and fitting a model is different from merging the traces of many UEs into a *single* trace and fitting the resulting inter-arrival time.



(a) CONNECTED (b) IDLE (c) HO (d) TAU

Figure 3: Variance-time plots for the CONNECTED and IDLE states and the HO and TAU events for phones.



(a) CONNECTED (b) IDLE (c) HO (d) TAU

Figure 4: Comparison of CDFs between real and fitted data (Poisson) for the CONNECTED and IDLE states and the HO and TAU events for phones.

change between CONNECTED and IDLE, their inter-arrival time still has stronger burstiness than the fitted Poisson models across the time scale from 10 to 10^3 seconds. The differences in the log-scale normalized variance between the real-world trace and the fitted exponential distribution are 0.20~1.20 and -0.04~0.63 for HO and TAU, respectively. We observe similar results for the other clusters and the other device types.

Inter-arrival tails. A more direct way of understanding why a traditional probability distribution cannot model a traffic trace is to examine whether the entire range of the observed inter-arrival time in the trace can be captured by the Poisson model [59]. We compare the CDF of the observed data with that of the fitted Poisson (exponential) distribution—over the same 1-hour interval for the sojourn time in CONNECTED and IDLE and the inter-arrival time of HO and TAU.

Figure 4 shows that for the randomly sampled cluster of phones, *the exponential distribution fails to adequately capture the entire range of the four quantities observed in the trace.* In particular, for CONNECTED, the maximum sojourn time is around 2106.94 seconds, much higher than that of the fitted exponential distribution, i.e., 156.35 seconds. For IDLE, the minimum sojourn time is around 0.16 seconds, smaller than that of the fitted exponential distribution, i.e., 0.40 seconds. For HO, the inter-arrival duration ranges from 0.69 to 1988.18 seconds, while the fitted inter-arrival duration only ranges from 0.78 to 559.56 seconds. As with HO, TAU has inter-arrival time ranging from 0.62 to 2721.36 seconds, while the fitted inter-arrival time only varies from 2.71 to 723.26 seconds. We observe similar results for all the other clusters.

5 MODELING FOR LTE

Our analysis above reveals three key challenges in modeling the control-plane of cellular networks: (**C1**) How to capture the intricate event dependence for each UE? (**C2**) How to derive control-plane traffic model to accurately capture the inter-arrival time between events? (**C3**) How to capture the significant diversity of control-plane traffic across UEs?

In this section, we propose *a two-level state-machine-based traffic model for each UE cluster (derived from an adaptive clustering scheme)* that addresses the three challenges above: (1) We develop a two-level state machine to capture the dependence among events for individual UEs; (2) We adopt the Semi-Markov Model to model the duration of a UE staying in the current state and the probability of transitioning to the next state; (3) We propose an adaptive clustering scheme to deal with the traffic diversity across UEs for each hour and for each device-type.

5.1 How to capture event dependence?

We observe that the six types of control events actually fall into two categories that have different dependence on each other; those that trigger a UE to transition among different UE states (denoted as Category-1 events), and those that do not but have complex dependence on UE states (denoted as Category-2 events).

Category-1 events include ATCH, DTCH, SRV_REQ, and S1_CONN_REL, which cause the UE state to switch from one state to another following the EMM and ECM state machines. We observe that when a UE changes from DEREGISTERED to REGISTERED, it always enters CONNECTED at the same time, which follows the 3GPP protocol [2]. Therefore, the EMM and ECM state machines can be merged as one state machine that captures the dependence (**C1**) of all Category-1 events, as shown in the top level in Figure 5. We denote it as the *EMM–ECM state machine.*

In contrast with Category-1 events, Category-2 events (HO and TAU) do not change the UE state. However, these events still have intricate dependence on other control events as follows. (1) For HO, it only happens after the UE enters CONNECTED triggered by SRV_REQ. (2) For TAU, although it can happen in both CONNECTED and IDLE, it sometimes follows HO in CONNECTED. This is because when the

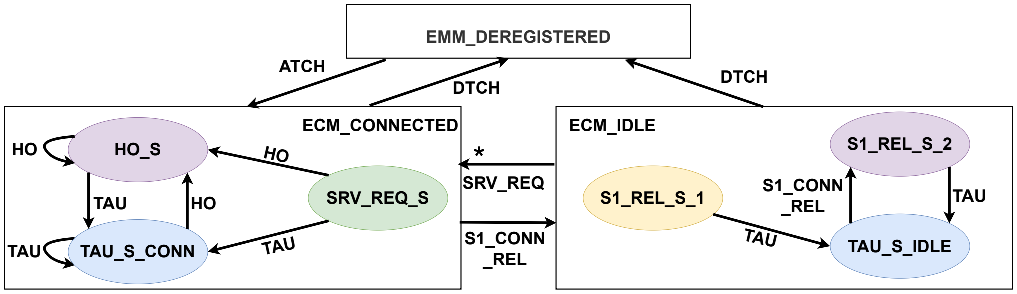


Figure 5: Proposed two-level state machine. The rectangles represent states at the top level. The ovals represent states at the bottom level. The arrow with the star denotes the SRV_REQ transition can only start from S1_REL_S_1 and S1_REL_S_2. The opposite arrow denotes the S1_CONN_REL transition can start from any of the states in CONNECTED.

UE moves and switches from one cell to another cell, it can also enter another tracking area (a new set of cells). However, TAU does not necessarily follow HO in CONNECTED, for example, when the UE reselects to LTE from 3G or the UE re-registers to LTE after the fallback for circuit-switched services, etc. (3) In the IDLE state, after TAU, S1_CONN_REL always happens in order to release the signaling resources assigned for the last TAU. None of these dependencies can be easily captured by the EMM–ECM state machine. The key challenge here is that these events can happen in either CONNECTED or IDLE or both, and with different dependencies on other events.

To capture the dependence (**C1**) for Category-2 events under the CONNECTED and IDLE states, we introduce two fine-grained sub-state machines that are embedded inside the CONNECTED and IDLE states of the EMM–ECM state machine, which effectively refines the EMM–ECM state machine into a two-level hierarchical state machine. In each sub-state machine, each state corresponds to the event that happens right before entering this state, e.g., the SRV_REQ_S state is entered after a SRV_REQ event. Each edge corresponds to a Category-2 control event, just like each edge in the EMM–ECM state machine corresponds to a Category-1 event.

Specifically, we define six new states in the sub-state machines, including HO_S, TAU_S_CONN, TAU_S_IDLE, SRV_REQ_S, S1_REL_S_1, and S1_REL_S_2. TAU_S_CONN and TAU_S_IDLE are needed to distinguish the state entered after a TAU event in the CONNECTED and IDLE states of the EMM–ECM state machine, respectively. Unlike S1_REL_S_1, which represents the UE switching from CONNECTED to IDLE, S1_REL_S_2 is needed to capture the unique behavior of TAU in IDLE.

Figure 5 (bottom left) shows the sub-state machine inside CONNECTED. After SRV_REQ happens and changes the UE to the SRV_REQ_S state, the UE can either trigger HO to enter HO_S or trigger TAU to enter TAU_S_CONN. When the UE is in HO_S, there are two possible transitions. If the next event is HO, it causes the UE to self-loop back to HO_S. If the next event is TAU, it changes the UE to TAU_S_CONN. Similarly, when the UE is in the TAU_S_CONN state, the next event is either TAU, which self-loops the UE to TAU_S_CONN, or HO, which changes the UE to HO_S.

Figure 5 (bottom right) shows the other sub-state machine inside IDLE. After S1_CONN_REL happens and changes the UE to the S1_REL_S_1 state, the UE can trigger TAU and then enters TAU_S_IDLE. Then, S1_CONN_REL is triggered to make the UE enter the S1_REL_S_2 state. When another TAU happens, UE changes back to the TAU_S_IDLE state.

In essence, those two sub-state machines are the second-level refined state machines embedded in the EMM–ECM state machine. Further, they can run concurrently with the top-level EMM–ECM state machine. For example, suppose that the UE is moving. The UE may switch from CONNECTED to IDLE, if the UE does not transmit data for some seconds. However, in the meantime, regardless of whether the UE is in CONNECTED or IDLE, TAU can happen, as shown in Figure 5.

5.2 How to derive control-plane traffic model?

Two-level state-machine-based Semi-Markov model. The two-level state machine captures the dependence among control events generated by each UE. To derive a control-plane traffic model, we convert the 2-level state machine into a *Semi-Markov model* [82]. The Semi-Markov is a multi-state model for a continuous time stochastic process with state transitions. We argue that the Semi-Markov model is a natural fit for modeling the control-plane traffic of cellular networks, since it can capture the intricate event dependence for each UE without relying on pre-defined distributions for the sojourn time in the traffic, which is a limitation of the classic Markov model. Specifically, unlike a Markov model, a Semi-Markov model does not assume that the sojourn time in the same state follows the exponential distribution with a constant hazard rate, which, as shown in §4.1, is not applicable to mobile network control traffic. In particular, given the 2-level state machine consisting of states and transitions, the Semi-Markov model models (1) the probability of the state transition from state x to state y: $p_{xy} = \mathbb{P}(S_{i+1} = y \mid S_i = x)$, where S_i and S_{i+1} represent the states of two consecutive steps i and $i+1$, respectively; (2) the duration of staying in state x before switching to state y, as a random variable with the CDF: $F_{xy}(t) = \mathbb{P}(T_{i+1} - T_i \leq t \mid S_i = x,\ S_{i+1} = y)$, where T_i and T_{i+1} represent the time of the process switching to the state S_i and S_{i+1}, respectively.

Deriving model parameters. To model the sojourn time in one state before switching to another state, we first collect the sojourn time for the same transition across all UEs. Since traditional probability distributions fail to model the duration, not only for the EMM and ECM states (§4.1), but also for the new states in the two-level

Table 2: Mapping of primary control-plane event types between 4G (left) and 5G (right).

ATCH	REGISTER (Registration)
DTCH	DEREGISTER (Deregistration)
SRV_REQ	SRV_REQ (Service Request)
S1_CONN_REL	AN_REL (AN Release)
HO	HO (Handover)
TAU	–

state machine (refer to Appendix A.3 for additional details), we derive one CDF model for the sojourn time of each transition. To model the transition probability from one state to another, we count the numbers of transitions across all UEs and then calculate p_{xy} for every transition (each edge in the state machine). If there is only one outbound edge, p_{xy} of the corresponding state transition is 1.

5.3 How to capture traffic diversity?

As discussed in §4.1.1, the UEs of the same device type have diverse traffic for each type of control-plane events and the distributions of the UEs are highly skewed for all three types of devices. As a result, using a single model for all UEs of the same device type fails to reproduce such diversity, as the sojourn time and the transition probabilities of more active UEs dominate corresponding CDFs. However, developing one model for every single UE cannot work, because of the lack of enough data to model.

Ideally, we should model as many *similar* UEs as possible. However, it is challenging to achieve both a large number of UEs and high similarity among the UEs, because of the highly diverse and skewed control-plane traffic over all UEs for all three types of devices.

To strike a balance between those two goals, we propose *an adaptive clustering scheme* to recursively segregate the UEs into different clusters, until the UEs in the cluster have roughly similar traffic patterns or the number of UEs in the cluster is small enough. To quantify the similarity of UE traffic, we focus on two dominant events, SRV_REQ and S1_CONN_REL, which contribute to 84.1%∼93.0% of the total control events for all three types of devices. We extract two features for each event type to characterize the UE traffic: (1) number of control events; (2) standard deviation of the duration staying in the state (CONNECTED or IDLE).

The recursive adaptive scheme works as follows. It is first invoked for the complete feature space and all UEs are grouped into one cluster. At each invocation, it checks for one cluster if the features of UEs are similar (i.e., the difference between maximum and minimum value should be smaller than some threshold θ_f for every feature), or if the number of UEs is smaller than another threshold θ_n. If neither one is satisfied, the procedure cuts the current feature space into 4 equal-sized sub-feature spaces, and the UEs that fall into the same sub-feature space are grouped into a sub-cluster. Effectively, the recursively partitioned sub-feature space forms a quadtree, and the UEs in the sub-feature space that is not partitioned further are grouped into a final cluster used to model the sojourn time and the transition probabilities for different state transitions. We conduct a binary search across the entire feature space and experimentally find that a θ_f value of 5 for all features and a θ_n value of 1000 are sufficient to segregate the UEs in the input trace into groups of UEs with sufficiently dissimilar behavior.

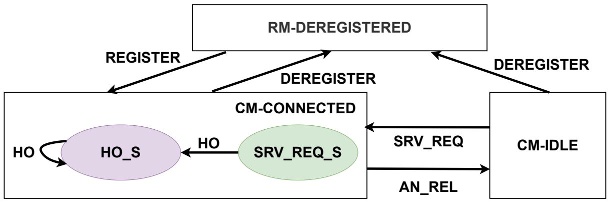


Figure 6: The adjusted two-level state machine for 5G.

Using the above thresholds, we generated 574, 199, and 70 UE clusters per hour on average for phones, connected cars, and tablets, respectively. As a result, we instantiated a total of 20,216 two-level state-machine-based Semi-Markov models, one for each combination of UE cluster, hour-of-day and device type.

5.4 Modeling the start event

Typical usage of the proposed traffic model is to generate a control-plane trace for a given number of UEs over a given duration. To achieve it, the traffic generator needs to decide the initial state for each UE. This means that the traffic generator needs to decide on the starting event and the corresponding start time for every UE, every time it synthesizes a new trace, e.g., given a starting hour of the day.

To do this, for each hour of the input trace, we collect the first event and the start time of all the UEs per UE-cluster per device-type discussed above, and derive the probabilities of different event types as the first event for the hour as well as the distribution of the start time within the hour.

6 MODELING FOR 5G

As mentioned in §3, one of the design goals of modeling control-plane traffic is to generate control traffic not only experienced by today's 4G MCN but also for future NextG (e.g., 5G). In this section, we present a methodology to showcase how to adapt LTE's two-level state machine[5] and how to derive model parameters for 5G. The reasons why we can easily adjust our proposed two-level state machine for LTE to for 5G are that (1) our proposed model is UE-centric (i.e. the generator generates control events between UE/RAN and MCN) and thus oblivious to the internal structural changes of MCN; (2) we observe that there exists a direct one-to-one mapping of all primary control-plane event types (except TAU) and UE states between 4G and 5G as discussed below.

Adjusting the two-level state machine. We carefully studied the 3GPP specifications of 4G [3] and 5G [4, 5]. We find that there exists a one-to-one mapping of event types between 4G and 5G for all primary types of control events, except TAU as summarized in Table 2. As for the four UE states in LTE (REGISTERED, DEREGISTERED, CONNECTED, and IDLE), there are also four corresponding UE states in 5G (RM-REGISTERED, RM-DEREGISTERED, CM-CONNECTED,

[5] The adjusted two-level state machine is applicable to 5G SA (standalone), while the two-level state machine for LTE is applicable to 5G NSA (non-standalone). This is because 5G NSA operates on LTE's MCN and thus LTE and 5G NSA share the same event types.

and CM-IDLE). Such high similarity of control-plane event types and UE states between 4G and 5G enables us to easily adjust the state machine of 4G to 5G, by simply removing the states and related transitions of TAU from 4G's state machine (Fig. 5) to form a new state machine for 5G (Fig. 6).

Deriving the model parameters. To instantiate the adjusted two-level state-machine-based traffic model for 5G, there are two ways: (1) When a large-scale 5G control-plane trace 5G is available, we just need to follow the same methodology (§5) and derive the model parameters. (2) When such a 5G trace is not available, e.g., due to privacy concerns, we can scale the parameters of the proposed traffic model for 4G to derive the model parameters for 5G.

Specifically, if we can estimate the scaling of each type of events when a UE switches from 4G to 5G, we can use the scaling factors to adjust the parameters in the 4G model for the 5G model. For example, recent measurement studies (e.g., [32]) have shown UEs tend to incur on average 4.6× more HO events when switching from 4G to 5G mmWave (NSA). Such frequency changes of some types of events can be used to recalculate the transition probabilities and the sojourn time in the two-level state machine for 5G (Fig. 6).

7 GENERATING TRACES FOR LTE & 5G

For LTE and 5G, the traffic generator uses the two-level state-machine-based traffic models to generate new control traffic traces for any given number of UEs, starting at any given hour, in the same following way.

To synthesize a new trace for K UEs starting at hour H, the main traffic generator runs K instances of the per-UE traffic generator concurrently. Each per-UE traffic generator uses the traffic models (for different hours) of a given UE cluster, according to the distribution of the UEs in the modeled trace, e.g., if 33% of the UEs belong to Cluster X, then 33% of the per-UE traffic generators will be running the state machine for Cluster X. The per-UE traffic generator first decides the first event and the corresponding start time within hour H by following the first-event model, and then starts driving the per-hour state machine for that UE cluster for that device type one hour after another hour.

In more detail, each per-UE traffic generator first samples the first event and the corresponding start time from the first-event model. It then runs the per-hour two-level state machine one after another, starting from hour H. For each level of the two-level state machine, it keeps a timer to track the time the UE stays in the current state. Whenever the UE enters a state x, the traffic generator (1) follows the probabilities $\{p_{xy} \mid y \in \text{all states coming from } x\}$ to decide the next state y and the corresponding event e that triggers the state transition from x to y; and (2) follows the CDF (F_{xy}) to decide the duration D for staying in the current state x. The traffic generator sets the timer for D seconds and starts it. When the timer expires, the UE generates event e and enters state y, and the traffic generator repeats the process above. In addition, when the state at the top level is changed, then for the bottom level, the traffic generator (1) drops its next event, which was decided to happen later in the past top-level state, (2) resets its timer, and (3) starts running the sub-state machine corresponding to the new state of the top level, e.g., Figure 5 (bottom left) if the top-level state machine enters CONNECTED.

Table 3: Comparison of different modeling methods.

Method	Base	$\mathbb{V}1$	$\mathbb{V}2$	Ours
State Machine	EMM−ECM	EMM−ECM	2−level	2−level
Distribution	Poisson	Poisson	Poisson	CDF
UE Clustering	×	✓	✓	✓

8 VALIDATION & EVALUATION

8.1 Model Validation for LTE

For LTE, we validate the proposed two-level state-machine-based traffic model by comparing it with the baseline method. In addition, to study the contribution of individual features of our method, we also evaluate two *variations* of our method by varying one feature at a time. Table 3 compares those different modeling methods. Specifically, (1) the baseline adopts the EMM−ECM state machine, which only captures ATCH, DTCH, SRV_REQ, and S1_CONN_REL. For those four event types, it fits the sojourn time of all UEs of the same device type staying in DEREGSITERED, CONNECTED and IDLE to Poisson distributions, using MLE. For the other event types, HO and TAU, it fits the inter-arrival time of HO and TAU to Poisson. (2) $\mathbb{V}1$ adopts the EMM−ECM state machine. It applies the same UE clusters as our method for the three device types. For each UE cluster, it fits the inter-arrival/sojourn time for all six event types to Poisson distributions. (3) $\mathbb{V}2$ models the sojourn time for the states in our method as Poisson processes using MLE, instead of using CDFs.

To assess the scalability of our proposed method and the other methods, we consider two different validation scenarios with different numbers of UEs: Scenario 1 with 38,000 UEs and Scenario 2 with 380,000 UEs, i.e., about 1× and 10× more than the UEs that we used to estimate the model parameters. For each method, we synthesize two traces for those two scenarios for one of the busy hours on a randomly-chosen day in August 2022. We then compare the synthesized traces with the real traces that we randomly sampled for the corresponding numbers of UEs and hour of the day from the same mobile operator in the entire US as §4.

The traffic generator ran on the computer equipped with 12 Intel Xeon (R) E5-2609 v3 CPUs with the base frequency of 1.90 GHz. The traffic generator initiated 38K and 380K instances of the per-UE traffic generator for the two scenarios, respectively. These instances were then scheduled as individual jobs to run on the 12 CPUs in parallel using the parallel command with one job assigned to each CPU at any given time. On average, it took 1.46/0.68/0.55 seconds for the per-UE traffic generators to synthesize a one-hour trace per phone/connected car/tablet.

To evaluate the fidelity of the synthesized traces using the two methods, we consider metrics from two perspectives: *macroscopic* and *microscopic*. From the macroscopic perspective, we compare the breakdown of the synthesized events with that of the real trace for all UEs of each type of devices. From the microscopic perspective, we zoom into the events generated for each UE, and compare per-UE traffic behavior.

8.1.1 Macroscopic analysis.

We first compute the breakdown of the synthesized events and compare it with that of the real events for each type of devices. We find that Scenario 1 and Scenario 2

Table 4: Differences of breakdown of events between the real trace and the synthesized traces generated by different methods under Scenario 2 with 380K UEs. The smaller the differences, the more accurate the model.

	Phones					Connected Cars					Tablets				
	Real	Base	$\mathbb{V1}$	$\mathbb{V2}$	**Ours**	Real	Base	$\mathbb{V1}$	$\mathbb{V2}$	**Ours**	Real	Base	$\mathbb{V1}$	$\mathbb{V2}$	**Ours**
ATCH	0.1%	-0.1%	+0.1%	+0.0%	**+0.0%**	0.8%	-0.8%	-0.6%	+0.4%	**+0.3%**	0.5%	-0.5%	-0.2%	+0.5%	**+0.6%**
DTCH	2.0%	-2.0%	-1.6%	+0.2%	**+0.1%**	6.2%	-6.2%	-5.8%	+0.7%	**+0.6%**	2.3%	-2.2%	-1.7%	+0.8%	**+0.8%**
SRV_REQ	45.5%	-45.3%	-35.4%	+1.8%	**+1.4%**	38.2%	-38.1%	-33.9%	+5.0%	**+4.5%**	46.3%	-46.1%	-32.1%	+0.4%	**-0.0%**
S1_CONN_REL	46.8%	-46.7%	-36.5%	+1.3%	**+1.0%**	42.2%	-42.0%	-37.4%	+2.8%	**+2.5%**	47.7%	-47.5%	-33.0%	+0.1%	**-0.1%**
HO (CONN.)	3.5%	+45.4%	+19.9%	-2.1%	**-1.7%**	7.5%	+41.4%	+12.7%	-5.5%	**-4.9%**	1.9%	+47.3%	+10.1%	-1.0%	**-0.7%**
HO (IDLE)	0.0%	+47.8%	+35.0%	+0.0%	**+0.0%**	0.0%	+47.7%	+38.5%	+0.0%	**+0.0%**	0.0%	+47.5%	+21.7%	+0.0%	**+0.0%**
TAU (CONN.)	0.7%	+2.2%	+10.7%	-0.4%	**-0.3%**	1.3%	+1.6%	+15.1%	-0.9%	**-0.8%**	0.3%	+2.6%	+5.0%	-0.2%	**-0.1%**
TAU (IDLE)	1.5%	-1.3%	+7.9%	-0.7%	**-0.6%**	3.8%	-3.6%	+11.3%	-2.4%	**-2.2%**	1.1%	-1.0%	+30.1%	-0.5%	**-0.4%**

have very similar results. We next elaborate the results for Scenario 2. Additional results for Scenario 1 can be found in Appendix B.

Table 4 shows that for Scenario 2, our method synthesizes a trace whose breakdowns of events for all three types of devices are closer to those of the real trace, compared with those of the traces generated by the other methods. For SRV_REQ/S1_CONN_REL, the percentages in the synthesized trace by our method only differ from those in the real trace by 1.4%/1.0% (phones), 4.5%/2.5% (connected cars) and -0.0%/-0.1% (tablets). In contrast, for the baseline, the percentages of those two event types differ the most from those in the real trace by -45.3%/-46.7% (phones), -38.1%/-42.0% (connected cars) and -46.1%/-47.5% (tablets). With clustering, $\mathbb{V1}$ achieves smaller differences than the baseline, i.e., -35.4%/-36.5% (phones), -33.9%/-37.4% (connected cars) and -32.1%/-33.0% (tablets). Finally, $\mathbb{V2}$ achieves the differences of 1.8%/1.3% (phones), 5.0%/2.8% (connected cars) and 0.4%/0.1% (tablets), slightly higher than our method. This is because we use MLE to best fit the Poisson distributions for the states in the two-level state machine for each UE cluster and thus the total numbers of different synthesized events for all UEs of the same device type are similar to those of the real trace.

Table 4 also shows that for HO events, which should happen only in the CONNECTED state, our method reproduces similar fractions of HO in CONNECTED and stops generating HO in IDLE. Specifically, in CONNECTED, the absolute differences in the percentages between our synthesized trace and the real trace are 0.7%~4.9%, smaller than the other methods for the three device types. In IDLE, the baseline and $\mathbb{V1}$ mistakenly generate 21.7%~47.8% of the total events as HO for the three device types, because the state dependence for HO is not captured by the EMM–ECM state machine. For TAU, which can happen in both CONNECTED and IDLE, our method can synthesize TAU in different ECM states correctly. The absolute differences in the percentages between the synthesized and real traces for TAU in CONNECTED and IDLE are 0.1%~2.2% for the three device types. In contrast, the traces synthesized by the baseline and $\mathbb{V1}$, which follow the EMM–ECM state machine, have larger differences from the real trace, i.e., 1.0%~30.1%. The results above show that our proposed two-level state machine can capture the state dependence well.

The synthesized trace also preserves high-level trends of control events across different device types. For example, Table 4 shows that for the four dominant types of control-plane events (i.e., SRV_REQ, S1_CONN_REL, HO, and TAU), phones and tablets have 1.2×/1.1× larger percentages of SRV_REQ/ S1_CONN_REL events than connected cars. This result suggests that phone and tablet users have a higher frequency of sending/receiving data to/from the Internet than connected cars. Compared with phones and tablets, connected cars have 2.0×/2.6× and 4.0×/3.4× larger percentages of HO/TAU events respectively, due to their higher mobility.

8.1.2 Microscopic analysis.

We next evaluate the microscopic fidelity of the synthesized traces in two ways: (1) numbers of events per UE for different event types; (2) sojourn time of each UE staying in one state before transiting to another state for different state transitions. We compare the distributions between real and synthesized traces by first deriving the CDFs for both traces and then computing the maximum y-distance, i.e., the distance along the y-axis (the probability of any "event" less than the x-axis value) of the two CDFs, which is a conservative way to measure the fidelity of the synthesized trace.

To validate traditional distribution-based modeling (e.g., Poisson distributions) cannot properly model the inter-arrival/ sojourn time for individual UE traffic, we compare the maximum y-distance between $\mathbb{V2}$ and our method for apples-to-apples comparisons. Since SRV_REQ and S1_CONN_REL dominate the total control events (i.e., 84.1%~93.0% of the total events shown in Table 1), we focus on those two events and the corresponding UE states, CONNECTED and IDLE.

Number of events per UE. Table 5 shows that for phones, our traffic generator can synthesize similar numbers of SRV_- REQ and S1_CONN_REL per UE under both validation scenarios; the maximum y-distance between CDFs of the synthesized and real traces is 6.7%~7.0% using our method, while it is 52.1%~53.1% using $\mathbb{V2}$. However, for connected cars and tablets, SRV_REQ and S1_CONN_REL synthesized by our proposed model have the maximum y-distance of 32.0%~33.2% (connected cars) and 16.0%~17.2% (tablets). In contrast, the CDFs for SRV_REQ and S1_CONN_REL generated by $\mathbb{V2}$ have even larger maximum y-distance of 37.5%~38.8% (connected cars) and 52.3%~52.8% (tablets) from those in the real trace.

To understand the high maximum y-distance for connected cars and tablets, we zoom into the CDFs of the number of SRV_REQ/S1_-CONN_REL events per UE in the real and synthesized traces for all three types of devices (Appendix C). We find that for both SRV_REQ and S1_CONN_REL in the synthesized trace by our proposed traffic model, the high maximum y-distance of connected cars and tablets

Table 5: Maximum y-distance between CDFs of numbers of SRV_REQ/S1_CONN_REL per UE and sojourn time in CONNECTED/IDLE per UE for the synthesized and real traces. The smaller the y-distance, the more accurate the model.

	Scenario 1 (38K UEs)						**Scenario 2** (380K UEs)					
	Phones		**Conn. Cars**		**Tablets**		**Phones**		**Conn. Cars**		**Tablets**	
	$\mathbb{V}2$	Ours	$\mathbb{V}2$	Ours	$\mathbb{V}2$	Ours	$\mathbb{V}2$	Ours	$\mathbb{V}2$	Ours	$\mathbb{V}2$	Ours
SRV_REQ	53.1%	6.9%	38.2%	33.2%	52.8%	16.7%	52.8%	6.7%	37.5%	32.3%	52.5%	16.0%
S1_CONN_REL	52.4%	7.0%	38.8%	32.9%	52.6%	17.2%	52.1%	6.8%	37.9%	32.0%	52.3%	17.0%
CONNECTED	30.2%	6.3%	25.0%	9.4%	23.4%	2.7%	31.0%	6.1%	23.5%	6.5%	23.1%	2.1%
IDLE	15.5%	4.8%	14.4%	11.7%	23.0%	8.2%	15.2%	4.3%	13.7%	10.4%	21.7%	6.8%

Table 6: Maximum y-distance between CDFs of numbers of events per UE for the synthesized and real traces for inactive / active UE groups per device type. The smaller the y-distance, the more accurate the model.

	Scenario 1 (38K UEs)		**Scenario 2** (380K UEs)	
	Conn. Cars	**Tablets**	**Conn. Cars**	**Tablets**
SRV_REQ	24.7%/12.2%	20.7%/9.8%	25.3%/11.1%	22.7%/7.8%
S1_CONN_REL	23.1%/11.8%	28.4%/9.9%	22.8%/10.6%	30.8%/7.6%

is caused by the UEs that generate only 1 event occurrence during the selected hour, while the traffic generator predicts 2 occurrences.

To illustrate the finding above, we focus on our model. For each type of devices, we split the UEs into two groups: (1) inactive UEs with fewer than or equal to 2 occurrences; (2) active UEs with more than 2 occurrences during the selected hour. We calculate the maximum y-distance between the CDFs of the synthesized and real traces for the two groups of UEs separately, for each device type.

Table 6 shows that for synthesized SRV_REQ and S1_CONN_REL using our proposed model, the maximum y-distance for active UEs with more than 2 events per UE is 10.6%~12.2% (connected cars) and 7.6%~9.9% (tablets) under the two validation scenarios, while the maximum y-distance for inactive UEs with fewer than or equal to 2 events per UE is 22.8%~25.3% (connected cars) and 20.7%~30.8% (tablets) under the two validation scenarios. The results suggest our proposed traffic model only mis-predicts the number of events by 1 during the selected hour for connected cars and tablets. We argue that such a difference by just 1 event per UE within a single hour for inactive connected cars and tablets is acceptable.

Duration of staying in one state before switching to the next state per UE. Table 5 also shows that our proposed method can simulate more accurate distributions of the sojourn time than $\mathbb{V}2$ for CONNECTED and IDLE states; the maximum y-distance for our method for both states is 2.1%~11.7% under both validation scenarios for the three device types, while it is 13.7%~31.0% for $\mathbb{V}2$.

8.2 Model Evaluation for 5G

For 5G, we evaluate the adjusted state-machine-based traffic models by considering two different deployment modes of 5G: non-standalone (NSA) and standalone (SA) using the mmWave frequency band. 5G NSA integrates 5G RAN with existing LTE RAN and MCN, while 5G SA operates independently with 5G MCN. Therefore, we use the two-level state machine of LTE (Fig. 5) to capture the event dependence for 5G NSA and the state machine of 5G (Fig. 6) for 5G SA. To derive the model parameters, we scale the parameters of 4G's traffic model for 5G NSA and 5G SA, respectively, since systematic extensive trace collection is not yet available for 5G. We focus on HO events as HO is significantly affected by 5G deployments, especially considering 5G mmWave base stations offer much smaller coverage areas compared to LTE. For 5G NSA, we use the scaling factor of 4.6× for HO (following [32]). To estimate the scaling factor for 5G SA, we perform a controlled experiment by collecting HO events of LTE and 5G mmWave on two phones (connecting to LTE and 5G mmWave, separately) under the same walking/driving scenarios. We calculate the ratio of number of HO events under 5G mmWave over LTE as the scaling factor, i.e., 3.0×. We then derive the model parameters for 5G NSA and 5G SA respectively (§6). Finally, we synthesize a 7-day trace for the same set of phones, connected cars, and tablets as the LTE trace mentioned in §4.

Table 7: Projected breakdown of control-plane events of 5G NSA and 5G SA for different types of devices.

Event Type	**P**		**CC**		**T**	
(NSA/SA)	NSA	SA	NSA	SA	NSA	SA
ATCH/REGISTER	0.1%	0.1%	0.8%	0.9%	1.1%	1.2%
DTCH/DEREGISTER	0.1%	0.2%	0.7%	0.9%	1.0%	1.1%
SRV_REQ/SRV_REQ	41.7%	45.3%	36.4%	42.7%	44.4%	47.6%
S1_CONN_REL/AN_REL	40.1%	43.5%	31.4%	36.8%	40.8%	43.8%
HO/HO	15.4%	10.9%	24.7%	18.8%	9.1%	6.4%
TAU/−	2.5%	−	6.0%	−	3.7%	−

Table 7 shows that compared with LTE, the percentage of HO events under 5G NSA/SA significantly increases from 3.8% to 15.4%/10.9% for phones, from 6.6% to 24.7%/18.8% for connected cars, and from 2.1% to 9.1%/6.4%. 5G NSA has more HO events than 5G SA, because in 5G NSA HO happens for not only 5G RAN but also LTE RAN.

9 DISCUSSION: GENERALIZABILITY

We have presented the methodology of modeling and generating control-plane traffic of cellular networks, which meet the four key requirements listed in §3. While the parameters of the proposed model are based on an extensive collection of real UEs of a major mobile carrier in the US, there are some factors that may cause changes to the control-plane traffic characteristics and thus the model parameters. In addition to change of cellular radio technologies (discussed in §6), carriers in distinct geographic regions (e.g., the US and countries in Asia) may deploy base stations diversely, people in distinct geographic regions may also use their devices differently, and new types of devices or applications (e.g., massive IoT, self-driving cars, etc.) may have different control-plane traffic patterns from the three device types studied in this paper, all of which could change the control-plane traffic characteristics. We believe that our modeling methodology is applicable to derive corresponding model parameters for cellular networks with such different characteristics.

10 RELATED WORK

LTE/5G traffic modeling. Most of previous work on traffic modeling for LTE and 5G [8, 12, 18–20, 24, 28, 33, 36, 37, 41, 42, 44, 45, 48, 51, 53, 55, 61, 62, 70–72, 74, 75, 83–87] focuses on the data-plane traffic, instead of the control-plane traffic. For the control-plane traffic, Dababneh et al. [24] proposed to model the total control-plane volume on different functions of LTE's MCN, using the number of UEs and events' transactions per second per subscriber. However, they ignored the traffic diversity in device types and time-of-day, and they also did not model the fine-grained inter-arrival time of successive events for individual UEs.

Internet traffic modeling. There has been extensive literature on modeling Internet traffic. Various probability distributions have been utilized to model the inter-arrival time of Internet traffic, including Poisson [15, 30, 40, 59], Pareto [6, 30], Weibull [11], Tcplib [30], etc. However, traditional probability distributions fail to model the control-plane traffic of cellular networks (§4) with proper state dependence per UE (§8.1). Recently, a variety of studies have been leveraging machine learning techniques to model temporal properties of Internet traffic (e.g., [47, 64, 65, 68, 78, 80]). However, it remains unclear whether their models can capture fine-grained temporal properties (e.g., distribution of inter-arrival time of events) and state dependence of the control-plane traffic of cellular networks. We leave it as future work.

LTE/5G control-plane traffic characterization. Existing works on characterizing control-plane traffic of cellular networks focus only on specific event types, e.g., HO [27, 32, 46] and ATCH and DTCH [73]. In this work, we perform an in-depth characterization study of the control-plane traffic of real UEs in the US (encompassing three prominent types of devices) for the six primary event types in LTE. Since 5G deployment is still in its early stage, we leave the study of large-scale 5G control-plane traffic as future work.

11 CONCLUSION

Accurate modeling and generation of control-plane traffic in today's and future mobile networks has important applications such as evaluating and optimizing MCN design and real-time monitoring of production MCNs. In this paper, we presented a two-level hierarchical state-machine-based control-plane traffic model based on the Semi-Markov Model that can accurately model per-UE control-plane traffic in cellular networks for arbitrary UE populations. Our validation shows that our model outperforms traditional probability-based model and can synthesize realistic traces for a much larger UE population than the trace used to instantiate the model. Specifically, compared with the real traces, our synthesized traces achieve small differences, i.e., within 1.7%, 4.9% and 0.8%, for phones, connected cars, and tablets, respectively. The developed traffic models in this paper are already being actively used by the Aether community to study the scalability of Aether 5G core design. We have open-sourced the developed traffic models to the wider community to stimulate research on MCN control-plane design and optimization for 4G/5G and beyond.

ACKNOWLEDGMENTS

We thank the anonymous reviewers and our shepherd Mallesham Dasari for their helpful comments. This work is supported in part by Intel.

REFERENCES

[1] 3GPP. 2016. *Architecture Enhancements for Control and User Plane Separation of EPC Nodes.* Technical Specification (TS) 23.214. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3077

[2] 3GPP. 2016. *General Packet Radio Service (GPRS) Enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Access.* Technical Specification (TS) 23.401. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=849

[3] 3GPP. 2016. *Non-Access-Stratum (NAS) Protocol for Evolved Packet System (EPS); Stage 3.* Technical Specification (TS) 24.301. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1072

[4] 3GPP. 2016. *System Architecture for the 5G System.* Technical Specification (TS) 23.501. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144

[5] 3GPP. 2021. *Procedures for the 5G System (5GS).* Technical Specification (TS) 23.502. https://www.etsi.org/deliver/etsi_ts/123500_123599/123502/15.15.00_60/ts_123502v151500p.pdf

[6] Abdelnaser Adas. 1997. Traffic Models in Broadband Networks. *IEEE Communications Magazine* (1997).

[7] Anup Agarwal, Zaoxing Liu, and Srinivasan Seshan. 2022. HeteroSketch: Coordinating Network-Wide Monitoring in Heterogeneous and Dynamic Networks. In *USENIX NSDI.*

[8] Alberto Martínez Alba and Wolfgang Kellerer. 2019. Large-and Small-Scale Modeling of User Traffic in 5G Networks. In *IEEE CNSM.*

[9] Strategy Analytics. 2021. 5G Signaling and Control Plane Traffic Depends on Service Communications Proxy (SCP). https://carrier.huawei.com/~/media/cnbgv2/download/products/core/strategy-analytics-5g-signaling-en.pdf.

[10] Anonymity. 2023. Paper under Submission.

[11] Muhammad Asad Arfeen, Krzysztof Pawlikowski, Don McNickle, and Andreas Willig. 2013. The Role of the Weibull Distribution in Internet Traffic Modeling. In *IEEE ITC.*

[12] Ayari Aymen. 2021. New Traffic Modeling for IoV/V2X in 5G Network Based on Data Mining. In *IEEE VTC2021-Spring.*

[13] Arijit Banerjee, Rajesh Mahindra, Karthik Sundaresan, Sneha Kasera, Kobus Van der Merwe, and Sampath Rangarajan. 2015. Scaling the LTE Control-Plane for Future Mobile Access. In *ACM CoNEXT.*

[14] Abhik Bose, Shailendra Kirtikar, Shivaji Chirumamilla, Rinku Shah, and Mythili Vutukuru. 2022. AccelUPF: Accelerating the 5G User Plane Using Programmable Hardware. In *ACM SOSR.*

[15] Jin Cao, William S Cleveland, Dong Lin, and Don X Sun. 2003. Internet Traffic Tends toward Poisson and Independent as the Load Increases. In *Nonlinear estimation and classification.* Springer.

[16] Indra Mohan Chakravarti, Radha G Laha, and Jogabrata Roy. 1967. Handbook of Methods of Applied Statistics. *Wiley Series in Probability and Mathematical Statistics (USA) eng* (1967).

[17] Moses Charikar, Kevin Chen, and Martin Farach-Colton. 2002. Finding Frequent Items in Data Streams. In *International Colloquium on Automata, Languages, and Programming.* Springer.

[18] Aleksandra Checko, Lars Ellegaard, and Michael Berger. 2012. Capacity Planning for Carrier Ethernet LTE Backhaul Networks. In *IEEE WCNC.*

[19] Min Chen, Yiming Miao, Hamid Gharavi, Long Hu, and Iztok Humar. 2019. Intelligent Traffic Adaptive Resource Allocation for Edge Computing-Based 5G Networks. *IEEE TCCN* (2019).

[20] Mingzi Chen, Xin Wei, Yun Gao, Liqi Huang, Mingkai Chen, and Bin Kang. 2020. Deep-Broad Learning System for Traffic Flow Prediction toward 5G Cellular Wireless Network. In *IEEE IWCMC.*

[21] Erhan Çinlar. 1968. On the Superposition of M-Dimensional Point Processes. *Journal of Applied Probability* (1968).

[22] B. Claise. 2004. Cisco Systems NetFlow Services Export Version 9.

[23] Graham Cormode. 2008. Count-Min Sketch. *Encyclopedia of Algorithms* (2008).

[24] Dima Dababneh, Marc St-Hilaire, and Christian Makaya. 2015. Data and Control Plane Traffic Modelling for LTE Networks. *Mobile Networks and Applications* (2015).

[25] Peter Danzig, Sugih Jamin, Ramón Cáceres, D Mitzel, and Deborah Estrin. 1992. An Empirical Workload Model for Driving Wide-Area TCP/IP Network Simulations. *Internetworking: Research and Experience* (1992).

[26] Peter B Danzig and Sugih Jamin. 1991. Tcplib: A Library of Internetwork Traffic Characteristics. (1991).

[27] Haotian Deng, Chunyi Peng, Ans Fida, Jiayi Meng, and Y Charlie Hu. 2018. Mobility Support in Cellular Networks: A Measurement Study on Its Configurations and Implications. In *ACM IMC*.

[28] Venkata Subbaraju Dommaraju, Karthik Nathani, Usman Tariq, Fadi Al-Turjman, Suresh Kallam, Rizwan Patan, et al. 2020. ECMCRR-MPDNL for Cellular Network Traffic Prediction with Big Data. *IEEE Access* (2020).

[29] Ericsson. 2021. Ericsson Mobility Report. https://www.ericsson.com/4ad7e9/assets/local/reports-papers/mobility-report/documents/2021/ericsson-mobility-report-november-2021.pdf.

[30] Victor S Frost and Benjamin Melamed. 1994. Traffic Modeling for Telecommunications Networks. *IEEE Communications Magazine* (1994).

[31] Mark W Garrett and Walter Willinger. 1994. Analysis, Modeling and Generation of Self-Similar VBR Video Traffic. *ACM SIGCOMM Computer Communication Review* (1994).

[32] Ahmad Hassan, Arvind Narayanan, Anlan Zhang, Wei Ye, Ruiyang Zhu, Shuowei Jin, Jason Carpenter, Z Morley Mao, Feng Qian, and Zhi-Li Zhang. 2022. Vivisecting Mobility Management in 5G Cellular Networks. In *ACM SIGCOMM*.

[33] Kaiwen He, Xu Chen, Qiong Wu, Shuai Yu, and Zhi Zhou. 2020. Graph Attention Spatial-Temporal Network with Collaborative Global-Local Learning for Citywide Mobile Traffic Prediction. *IEEE TMC* (2020).

[34] Thomas P Hettmansperger and Michael A Keenan. 1975. Tailweight, Statistical Inference and Families of Distributions—a Brief Survey. *A Modern Course on Statistical Distributions in Scientific Work* (1975).

[35] Qun Huang, Xin Jin, Patrick PC Lee, Runhui Li, Lu Tang, Yi-Chao Chen, and Gong Zhang. 2017. Sketchvisor: Robust Network Measurement for Software Packet Processing. In *ACM SIGCOMM*.

[36] Yupin Huang, Liping Qian, Anqi Feng, Ningning Yu, and Yuan Wu. 2019. Short-Term Traffic Prediction by Two-Level Data Driven Model in 5G-Enabled Edge Computing Networks. *IEEE Access* (2019).

[37] Elias Jailani, Muhamad Ibrahim, and Ruhani Ab Rahman. 2012. LTE Speech Traffic Estimation for Network Dimensioning. In *IEEE ISWTA*.

[38] Vivek Jain, Hao-Tse Chu, Shixiong Qi, Chia-An Lee, Hung-Cheng Chang, Cheng-Ying Hsieh, KK Ramakrishnan, and Jyh-Cheng Chen. 2022. L25GC: a Low Latency 5G Core Network Based on High-Performance NFV Platforms. In *ACM SIGCOMM*.

[39] Xin Jin, Li Erran Li, Laurent Vanbever, and Jennifer Rexford. 2013. Softcell: Scalable and Flexible Cellular Core Network Architecture. In *ACM CoNEXT*.

[40] Thomas Karagiannis, Mart Molle, Michalis Faloutsos, and Andre Broido. 2004. A Nonstationary Poisson View of Internet Traffic. In *IEEE INFOCOM*.

[41] Daegyeom Kim, Myeongjin Ko, Sunghyun Kim, Sungwoo Moon, Kyung-Yul Cheon, Seungkeun Park, Yunbae Kim, Hyungoo Yoon, and Yong-Hoon Choi. 2022. Design and Implementation of Traffic Generation Model and Spectrum Requirement Calculator for Private 5G Network. *IEEE Access* (2022).

[42] D Kutuzov, A Osovsky, O Stukach, and D Starov. 2021. Modeling of IIoT Traffic Processing by Intra-Chip NoC Routers of 5G/6G Networks. In *IEEE SIBCON*.

[43] Will E Leland, Murad S Taqqu, Walter Willinger, and Daniel V Wilson. 1994. On the Self-Similar Nature of Ethernet Traffic (Extended Version). *IEEE/ACM ToN* (1994).

[44] Xi Li, Umar Toseef, Thushara Weerawardane, Wojciech Bigos, Dominik Dulas, Carmelita Goerg, Andreas Timm-Giel, and Andreas Klug. 2010. Dimensioning of the LTE Access Transport Network for Elastic Internet Traffic. In *IEEE WiMob*.

[45] Xi Li, Umar Toseef, Thushara Weerawardane, Wojciech Bigos, Dominik Dulas, Carmelita Goerg, Andreas Timm-Giel, and Andreas Klug. 2010. Dimensioning of the LTE S1 Interface. In *IEEE WMNC*.

[46] Yuanjie Li, Qianru Li, Zhehui Zhang, Ghufran Baig, Lili Qiu, and Songwu Lu. 2020. Beyond 5G: Reliable Extreme Mobility Management. In *ACM SIGCOMM*.

[47] Zinan Lin, Alankar Jain, Chen Wang, Giulia Fanti, and Vyas Sekar. 2020. Using GANs for Sharing Networked Time Series Data: Challenges, Initial Promise, and Open Questions. In *ACM IMC*.

[48] Karl Lindberger. 1999. Balancing Quality of Service, Pricing and Utilisation in Multiservice Networks with Stream and Elastic Traffic. *Teletraffic Science and Engineering* (1999).

[49] Zaoxing Liu, Ran Ben-Basat, Gil Einziger, Yaron Kassner, Vladimir Braverman, Roy Friedman, and Vyas Sekar. 2019. Nitrosketch: Robust and General Sketch-Based Monitoring in Software Switches. In *ACM SIGCOMM*.

[50] Zaoxing Liu, Antonis Manousis, Gregory Vorsanger, Vyas Sekar, and Vladimir Braverman. 2016. One Sketch to Rule Them All: Rethinking Network Flow Monitoring with Univmon. In *ACM SIGCOMM*.

[51] Josip Lorincz, Zvonimir Klarin, and Dinko Begusic. 2021. Modeling and Analysis of Data and Coverage Energy Efficiency for Different Demographic areas in 5G networks. *IEEE Systems Journal* (2021).

[52] Robert MacDavid, Carmelo Cascone, Pingping Lin, Badhrinath Padmanabhan, Ajay Thakur, Larry Peterson, Jennifer Rexford, and Oguz Sunay. 2021. A P4-Based 5G User Plane Function. In *ACM SOSR*.

[53] Chitradeep Majumdar, Miguel Lopez-Benitez, and Shabbir N Merchant. 2020. Real Smart Home Data-Assisted Statistical Traffic Modeling for the Internet of Things. *IEEE Internet of Things Journal* (2020).

[54] Frank J Massey Jr. 1951. The Kolmogorov-Smirnov Test for Goodness of Fit. *Journal of the American statistical Association* (1951).

[55] Florian Metzger, Tobias Hoßfeld, André Bauer, Samuel Kounev, and Poul E Heegaard. 2019. Modeling of Aggregated IoT Traffic and Its Application to an IoT Cloud. *Proc. IEEE* (2019).

[56] Mehrdad Moradi, Wenfei Wu, Li Erran Li, and Zhuoqing Morley Mao. 2014. SoftMoW: Recursive and Reconfigurable Cellular WAN Architecture. In *ACM CoNEXT*.

[57] Nokia Siemens Networks. 2016. Signaling is Growing 50% Faster than Data Traffic. http://goo.gl/uwnRiO.

[58] Francesco Paolucci, Davide Scano, Filippo Cugini, Andrea Sgambelluri, Luca Valcarenghi, Carlo Cavazzoni, Giuseppe Ferraris, and Piero Castoldi. 2021. User Plane Function Offloading in P4 Switches for Enhanced 5G Mobile Edge Computing. In *IEEE DRCN*.

[59] Vern Paxson and Sally Floyd. 1995. Wide Area Traffic: the Failure of Poisson Modeling. *IEEE/ACM ToN* (1995).

[60] Peter Phaal, Sonia Panchen, and Neil McKee. 2001. RFC3176: InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks.

[61] Nicola Piovesan, Antonio De Domenico, Matteo Bernabé, David López-Pérez, Harvey Baohongqiang, Geng Xinli, Wang Xie, and Mérouane Debbah. 2021. Forecasting Mobile Traffic to Achieve Greener 5G Networks: When Machine Learning is Key. In *IEEE SPAWC*.

[62] Roopesh Kumar Polaganga and Qilian Liang. 2015. Self-Similarity and Modeling of LTE/LTE-A Data Traffic. *Measurement* (2015).

[63] Zafar Ayyub Qazi, Melvin Walls, Aurojit Panda, Vyas Sekar, Sylvia Ratnasamy, and Scott Shenker. 2017. A High Performance Packet Core for Next Generation Cellular Networks. In *ACM SIGCOMM*.

[64] Maria Rigaki and Sebastian Garcia. 2018. Bringing a GAN to a Knife-Fight: Adapting Malware Communication to Avoid Detection. In *IEEE SPW*.

[65] Markus Ring, Daniel Schlör, Dieter Landes, and Andreas Hotho. 2019. Flow-Based Network Traffic Generation Using Generative Adversarial Networks. *Computers & Security* (2019).

[66] Horst Rinne. 2008. *The Weibull Distribution: A Handbook*. Chapman and Hall/CRC.

[67] Fritz W Scholz and Michael A Stephens. 1987. K-Sample Anderson–Darling Tests. *J. Amer. Statist. Assoc.* (1987).

[68] Mustafizur R Shahid, Gregory Blanc, Houda Jmila, Zonghua Zhang, and Hervé Debar. 2020. Generative Deep Learning for Internet of Things Network Traffic Generation. In *IEEE PRDC*.

[69] Michael A Stephens. 1974. EDF Statistics for Goodness of Fit and Some Comparisons. *Journal of the American statistical Association* (1974).

[70] Hoang Duy Trinh, Nicola Bui, Joerg Widmer, Lorenza Giupponi, and Paolo Dini. 2017. Analysis and Modeling of Mobile Traffic using Real Traces. In *IEEE PIMRC*.

[71] Hoang Duy Trinh, Lorenza Giupponi, and Paolo Dini. 2018. Mobile Traffic Prediction from Raw Data using LSTM Networks. In *IEEE PIMRC*.

[72] Jing Wang, Jian Tang, Zhiyuan Xu, Yanzhi Wang, Guoliang Xue, Xing Zhang, and Dejun Yang. 2017. Spatiotemporal Modeling and Prediction in Cellular Networks: A Big Data Enabled Deep Learning Approach. In *IEEE INFOCOM*.

[73] Jing Wang, Wenli Zhou, Huan Wang, and Luying Chen. 2014. A Control-Plane Traffic Analysis Tool for LTE Network. In *2014 Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics*. IEEE.

[74] Shuo Wang, Xing Zhang, Jiaxin Zhang, Jian Feng, Wenbo Wang, and Ke Xin. 2015. An Approach for Spatial-Temporal Traffic Modeling in Mobile Cellular Networks. In *IEEE International Teletraffic Congress*.

[75] Zi Wang, Jia Hu, Geyong Min, Zhiwei Zhao, Zheng Chang, and Zhe Wang. 2022. Spatial-Temporal Cellular Traffic Prediction for 5G and Beyond: A Graph Neural Networks-Based Approach. *IEEE T-IINF* (2022).

[76] Wikipedia. 2022. p-value. https://en.wikipedia.org/wiki/P-value.

[77] Wikipedia. 2022. Type Allocation Code (TAC). https://en.wikipedia.org/wiki/Type_Allocation_Code.

[78] Shengzhe Xu, Manish Marwah, and Naren Ramakrishnan. 2020. Stan: Synthetic Network Traffic Generation Using Autoregressive Neural Models. *arXiv preprint arXiv:2009.12740* (2020).

[79] Tong Yang, Jie Jiang, Peng Liu, Qun Huang, Junzhi Gong, Yang Zhou, Rui Miao, Xiaoming Li, and Steve Uhlig. 2018. Elastic Sketch: Adaptive and Fast Network-Wide Measurements. In *ACM SIGCOMM*.

[80] Yucheng Yin, Zinan Lin, Minhao Jin, Giulia Fanti, and Vyas Sekar. 2022. Practical GAN-Based Synthetic IP Header Trace Generation Using Netshare. In *ACM SIGCOMM*.

[81] Minlan Yu, Lavanya Jose, and Rui Miao. 2013. Software Defined Traffic Measurement with OpenSketch. In *USENIX NSDI*.

[82] Shun-Zheng Yu. 2010. Hidden Semi-Markov Models. *Artificial intelligence* (2010).

[83] Qingtian Zeng, Qiang Sun, Geng Chen, Hua Duan, Chao Li, and Ge Song. 2020. Traffic Prediction of Wireless Cellular Networks Based on Deep Transfer Learning and Cross-Domain Data. *IEEE Access* (2020).

[84] Chaoyun Zhang and Paul Patras. 2018. Long-Term Mobile Traffic Forecasting using Deep Spatio-Temporal Neural Networks. In *ACM Mobihoc*.

[85] Dehai Zhang, Linan Liu, Cheng Xie, Bing Yang, and Qing Liu. 2020. Citywide Cellular Traffic Prediction Based on a Hybrid Spatiotemporal Network. *Algorithms* (2020).

[86] Shuai Zhao, Xiaopeng Jiang, Guy Jacobson, Rittwik Jana, Wen-Ling Hsu, Raif Rustamov, Manoop Talasila, Syed Anwar Aftab, Yi Chen, and Cristian Borcea. 2020. Cellular Network Traffic Prediction Incorporating Handover: A Graph Convolutional Approach. In *IEEE SECON*.

[87] Michael Zink, Kyoungwon Suh, Yu Gu, and Jim Kurose. 2009. Characteristics of Youtube Network Traffic at a Campus Network–Measurements, Models, and Implications. *Computer Networks* (2009).

APPENDICES

A LIMITATIONS OF TRADITIONAL PROBABILITY DISTRIBUTIONS

In this appendix, we present additional results to show how traditional probability distributions fail to model the inter-arrival time between events, and the sojourn time in the four EMM/ECM states without leveraging our adaptive clustering scheme (Appendix A.1) and with leveraging our adaptive clustering scheme (Appendix A.2), as well as the new states in the proposed traffic model (Appendix A.3), for individual UE traffic.

A.1 Can Other Traditional Probability Distributions Model Individual UE Traffic without UE Clustering?

We first discuss the results showing that without leveraging our adaptive clustering scheme, the individual UE traffic (inter-arrival time of different types of events and the sojourn time staying in the four EMM/ECM states) cannot be fitted using the other traditional probability models, in addition to the Poisson model discussed in §4.1.

Results. Table 8 shows that neither the inter-arrival time nor the sojourn time can be modeled by Pareto, Weibull, and Tcplib distributions for each UE cluster of all three types of devices, with close to 0% of the 1-hour intervals that pass the K–S test for the Weibull/Pareto/Tcplib distributions.

A.2 Can Traditional Probability Distributions Model Individual UE Traffic with UE Clustering?

We next discuss the results showing the individual UE traffic (inter-arrival time of different types of events and the sojourn time staying in the four EMM/ECM states) cannot be fitted using traditional probability models after applying our adaptive clustering scheme.

Results. Table 9 shows that surprisingly the inter-arrival time of all six types of events (`ATCH`, `DTCH`, `SRV_REQ`, `S1_CONN- _REL`, `HO`, and `TAU`) cannot be modeled as Poisson processes for each UE cluster of all three types of devices, since below 5.0% of the 1-hour intervals pass the K–S test and below 23.8% for the A^2 test for the exponential distributions. For the duration of UE staying in the four EMM/ECM states (`REGISTERED`, `DEREGISTERED`, `CONNECTED`, and `IDLE`), below 1.4% of the intervals pass the two statistical tests for the exponential distributions for all three types of devices.

Table 9 also shows neither the inter-arrival time nor the sojourn time can be modeled by Pareto, Weibull, and Tcplib distributions for each UE cluster of all three types of devices. In particular, the Weibull distribution models achieve the largest percentage of the 1-hour intervals that pass the K-S test over all UE clusters, i.e., up to 40.0%, while the Pareto distributions and the Tcplib distributions have at most 10.2% and 1.5% of intervals that pass the K–S test.

A.3 Can Traditional Probability Distributions Model the Sojoun Time in New States Proposed in Our Model?

To capture the state dependence among events, we propose a two-level state machine where the first-level state machine is the EMM–ECM state machine and the second-level state machine includes six new states and nine corresponding state transitions as shown in Fig. 5. We next discuss whether those new states can be modeled using traditional probability distributions.

Following the same methodology mentioned in §4.1 we first apply traditional probability distributions to the sojourn time of each state for each combination of UE clusters, 1-hour intervals, and device types. We then apply both the K–S test and the A^2 test to the Poisson distribution and apply only the K–S test to the other distributions, since the A^2 test can only be applied to some common distributions at the moment (e.g., normal and exponential).

Results. Table 10 shows all traditional distributions cannot properly model the sojourn time of UE staying in those states for each UE cluster of all three types of devices. For the Poisson distribution, close to zero intervals and up to 2.9% of the intervals can pass the K–S test and the A^2 test, respectively. For the other distributions, the Pareto distribution achieves the largest percentage of the 1-hour intervals that pass the K-S test over all UE clusters, i.e., up to 24.5%, while the Weibull distribution and the Tcplib distribution have at most 21.5% and 3.3% of intervals that pass the test.

B SUPPLEMENTARY MACROSCOPIC ANALYSIS FOR MODEL VALIDATION

This appendix provides supplementary macroscopic analysis (discussed in §8.1) to validate the proposed models under Scenario 1 with 38K UEs.

Table 11 shows that for Scenario 1, the proposed method synthesizes a trace whose breakdowns of control-plane events for all three types of devices are much closer to those of the real trace, than those of the trace generated by all three baselines. For `SRV_REQ` and `S1_CONN_REL`, the percentages in the synthesized trace by our method only differ from those in the real trace by 1.3%/1.1% (phones), 5.0%/2.1% (connected cars) and 0.1%/-0.3% (tablets) . In contrast, for the baseline, the percentages of those two event types differ the most from those in the real trace by -45.5%/-46.6% (phones), -37.5%/-42.3% (connected cars) and -45.9%/-47.6% (tablets). With clustering, 𝕍𝟙 has smaller differences than the baseline, i.e., -36.4%/-37.2% (phones), -33.0%/-37.5% (connected cars), and -28.4%/-29.5% (tablets). As 𝕍𝟚 follows the proposed two-level state machine with clustering, the differences further decreases to 0.6%/0.3% (phones), 5.8%/2.7% (connected cars) and 0.5%/-0.1% (tablets).

Table 8: Percentages of the 1-hour intervals whose inter-arrival time of different event types or sojourn time in the four EMM/ECM states passes the statistical tests for traditional probability distributions without UE clustering.

Test	Device Type	ATCH	DTCH	SRV_REQ	S1_CONN_REL	HO	TAU	REG.	DEREG.	CONN.	IDLE
Poisson (K–S)	Phones	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Conn. Cars	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Tablets	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Poisson (A^2)	Phones	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Conn. Cars	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Tablets	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Pareto (K–S)	Phones	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Conn. Cars	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Tablets	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Weibull (K–S)	Phones	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Conn. Cars	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Tablets	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Tcplib (K–S)	Phones	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Conn. Cars	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Tablets	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

Table 9: Percentages of the 1-hour intervals whose inter-arrival time of different event types or sojourn time in the four EMM/ECM states passes the statistical tests for traditional probability distributions with UE clustering.

Test	Device Type	ATCH	DTCH	SRV_REQ	S1_CONN_REL	HO	TAU	REG.	DEREG.	CONN.	IDLE
Poisson (K–S)	Phones	2.0%	5.0%	0.5%	0.5%	0.1%	0.0%	0.0%	0.0%	0.0%	0.2%
	Conn. Cars	2.5%	5.0%	0.2%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Tablets	0.0%	0.6%	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.2%
Poisson (A^2)	Phones	3.0%	12.5%	3.0%	2.7%	0.2%	0.0%	0.0%	1.4%	0.1%	1.3%
	Conn. Cars	16.5%	23.8%	0.9%	0.0%	0.2%	0.1%	0.0%	0.0%	0.0%	0.1%
	Tablets	0.5%	6.5%	0.8%	0.2%	0.0%	0.0%	0.0%	0.0%	0.1%	1.0%
Pareto (K–S)	Phones	0.0%	0.0%	0.1%	0.7%	7.8%	10.2%	0.0%	0.0%	2.9%	5.0%
	Conn. Cars	0.0%	0.0%	0.0%	0.3%	3.9%	9.7%	0.0%	0.2%	0.0%	0.7%
	Tablets	0.5%	0.0%	0.5%	1.2%	5.0%	6.0%	0.0%	0.0%	3.4%	2.7%
Weibull (K–S)	Phones	22.0%	40.0%	7.5%	5.5%	10.2%	10.2%	0.0%	1.4%	1.1%	5.3%
	Conn. Cars	2.5%	0.0%	1.9%	0.8%	18.7%	10.4%	0.1%	0.0%	0.0%	0.5%
	Tablets	6.6%	20.7%	3.3%	3.3%	11.8%	7.3%	0.0%	3.4%	0.9%	1.7%
Tcplib (K–S)	Phones	0.0%	0.0%	0.4%	0.4%	0.3%	0.3%	0.0%	0.0%	0.3%	0.3%
	Conn. Cars	0.0%	0.0%	0.0%	0.0%	1.5%	1.5%	0.0%	0.0%	0.0%	0.0%
	Tablets	0.0%	0.0%	0.5%	0.6%	0.0%	0.0%	0.0%	0.0%	0.3%	0.3%

Table 11 also shows that for HO events, which should happen only in the CONNECTED state, our method reproduces similar fractions of HO in CONNECTED and stops generating HO in IDLE. Specifically, in CONNECTED, the absolute differences in the percentages between our synthesized trace and the real trace are -1.7% (phones), -4.6% (connected cars) and -0.3% (tablets). They are much smaller than those between the traces synthesized by the baseline and the real traces, i.e., 44.0% (phones), 40.9% (connected cars) and 46.7% (tablets). With clustering, $\mathbb{V}\mathbb{1}$ has much smaller differences from the real traces than Base, i.e., 21.3% (phones), 10.7% (connected cars) and 13.2% (tablets). After applying both clustering and the two-level state machine, $\mathbb{V}\mathbb{2}$ achieves the differences of -2.2% (phones), -5.3% (connected cars) and -0.7% (tablets). In IDLE, the baseline and $\mathbb{V}\mathbb{1}$ mistakenly generate 29.5%~46.6% of the total events as HO for the three device types, because the state dependence for HO is not captured by the EMM–ECM state machine. For TAU, which can happen in both CONNECTED and IDLE, our method can synthesize TAU in different ECM states correctly. The differences in the percentages between the synthesized and real traces for TAU in CONNECTED/IDLE are -0.3%/-0.5% (phones), -0.8%/-3.1% (connected cars), and -0.1%/-0.7% (tablets). In contrast, the traces synthesized by the baseline and $\mathbb{V}\mathbb{1}$, which follow the EMM–ECM state machine, have much larger differences from the real trace, i.e., 1.5%~10.8% (phones), -1.9%~15.9% (connected cars), and 1.3%~10.6% (tablets). The results above show that our proposed two-level state machine can capture the state dependence well.

C SUPPLEMENTARY MICROSCOPIC ANALYSIS FOR MODEL VALIDATION

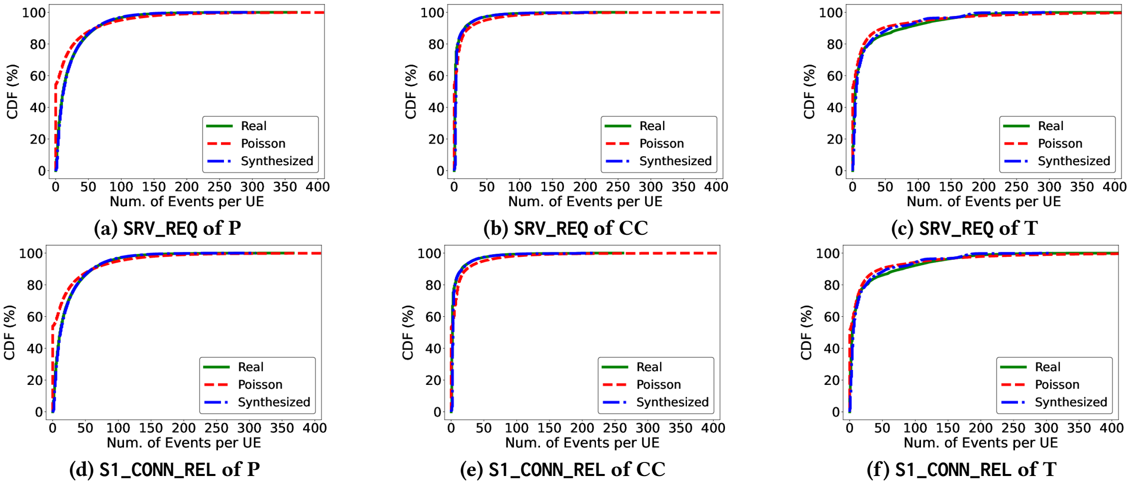
This appendix provides supplementary microscopic analysis (discussed in §8.1) to validate the proposed models. Specifically, for each device type, we plot the CDFs to examine the entire range of number of SRV_REQ/S1_CONN_REL per UE synthesized by our

Table 10: Percentages of 1-hour intervals over all clusters that pass the two standard statistical tests for the state transitions (denoted as outbound state–trigger event) in the two second-level state machines.

Test	Device Type	SRV_REQ_S-HO	HO_S-HO	TAU_S_C-HO	SRV_REQ_S-TAU	TAU_S_C-TAU	HO_S-TAU	S1_REL_1-TAU	S1_REL_2-TAU	TAU_S_I-S1_REL
Poisson (K–S)	Phones	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Conn. Cars	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Tablets	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Poisson (A^2)	Phones	0.0%	0.2%	0.9%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Conn. Cars	0.0%	0.1%	0.0%	0.0%	0.0%	2.9%	0.0%	0.0%	0.0%
	Tablets	0.0%	0.7%	0.0%	0.0%	0.0%	0.0%	0.8%	0.0%	0.0%
Pareto (K–S)	Phones	24.5%	5.9%	8.8%	0.0%	0.0%	0.0%	2.5%	1.6%	0.0%
	Conn. Cars	1.3%	4.8%	7.8%	0.0%	0.0%	1.3%	0.0%	0.3%	0.0%
	Tablets	14.2%	12.8%	10.0%	0.0%	0.0%	1.6%	0.0%	0.6%	0.0%
Weibull (K–S)	Phones	6.5%	5.2%	4.3%	0.0%	0.0%	0.4%	21.5%	12.3%	0.0%
	Conn. Cars	18.1%	15.6%	15.0%	0.0%	0.0%	17.0%	1.6%	0.9%	0.0%
	Tablets	16.7%	9.5%	3.3%	0.0%	0.0%	1.6%	9.0%	4.2%	0.0%
Tcplib (K–S)	Phones	0.5%	0.2%	1.2%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Conn. Cars	0.0%	0.4%	0.0%	0.0%	0.0%	0.0%	0.3%	0.0%	0.0%
	Tablets	0.0%	0.0%	3.3%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

Table 11: Differences of breakdown of events between the real trace and the synthesized traces generated by different methods for each device type under Scenario 1 with 38K UEs. The smaller the differences, the more accurate the model.

	Phones					Connected Cars					Tablets				
	Real	Base	𝕍𝟙	𝕍𝟚	Ours	Real	Base	𝕍𝟙	𝕍𝟚	Ours	Real	Base	𝕍𝟙	𝕍𝟚	Ours
ATCH	0.1%	-0.1%	+0.0%	+0.0%	+0.0%	0.8%	-0.8%	-0.6%	+0.5%	+0.4%	0.5%	-0.5%	-0.1%	+0.5%	+0.5%
DTCH	2.0%	-2.0%	-1.7%	+0.1%	+0.1%	6.1%	-6.1%	-5.7%	+1.0%	+1.0%	2.3%	-2.2%	-1.5%	+0.8%	+0.8%
SRV_REQ	45.6%	-45.5%	-36.4%	+0.6%	+1.3%	37.6%	-37.5%	-33.0%	+5.8%	+5.0%	46.1%	-45.9%	-28.4%	+0.5%	+0.1%
S1_CONN_REL	46.8%	-46.6%	-37.2%	+0.3%	+1.1%	42.4%	-42.3%	-37.5%	+2.7%	+2.1%	47.8%	-47.6%	-29.5%	-0.1%	-0.3%
HO (CONN.)	3.5%	+44.0%	+21.3%	-2.2%	-1.7%	7.1%	+40.9%	+10.7%	-5.3%	-4.6%	1.6%	+46.7%	+13.2%	-0.7%	-0.3%
HO (IDLE)	+0.0%	+46.6%	+32.6%	+0.0%	+0.0%	+0.0%	+46.1%	+38.3%	+0.0%	+0.0%	+0.0%	+45.8%	+29.5%	+0.0%	+0.0%
TAU (CONN.)	0.7%	+2.1%	+10.8%	-0.4%	-0.3%	1.3%	+1.5%	+15.9%	-0.9%	-0.8%	0.3%	+2.5%	+6.4%	-0.2%	-0.1%
TAU (IDLE)	1.2%	+1.5%	+10.5%	-0.6%	-0.5%	4.7%	-1.9%	+11.9%	-3.3%	-3.1%	1.5%	+1.3%	+10.6%	-0.9%	-0.7%



(a) SRV_REQ of P

(b) SRV_REQ of CC

(c) SRV_REQ of T

(d) S1_CONN_REL of P

(e) S1_CONN_REL of CC

(f) S1_CONN_REL of T

Figure 7: Comparison of CDFs of number of SRV_REQ/S1_CONN_REL per UE between the synthesized and real 1-hour traces for three types of devices in Scenario 2 with 380,000 UEs.

proposed method, and compare them with those generated by the baseline method.

Figure 7 presents under Scenario 2 with 38K UEs, similar to phones, both connected cars and tablets have no explicit visual difference in the y-axis of the CDFs between the real and synthesized traces using our proposed traffic model. However, for the baseline, there exist visible differences in the y-axis of the CDFs, compared with the real trace. Specifically, our proposed method has 3.52×∼7.92×, 1.16×∼3.63×, and 3.07×∼11.14× smaller maximum y-distance than the baseline method for phones, connected cars, and tablets, respectively.

D ETHICS

The original control-plane traffic traces collected have the user identity anonymized, ensuring the protection of user privacy. The anonymization process separates any personally identifiable information from the collected data, thereby preserving the anonymity of individuals. This work does not raise any ethical issues.