Published: 10 July 2025
Online AM: 29 May 2025
Accepted: 05 May 2025
Revised: 03 April 2025
Received: 31 December 2024

RESEARCH-ARTICLE

Sensing Resource Scheduling in 5G vRAN: An Elastic Approach

YULONG CHEN, Tsinghua University, Beijing, China

JUNCHEN GUO, Alibaba Group Holding Limited, Hangzhou, Zhejiang, China

YIMIAO SUN, Tsinghua University, Beijing, China

HAIPENG YAO, Beijing University of Posts and Telecommunications, Beijing, Beijing, China

YUNHAO LIU, Tsinghua University, Beijing, China

YUAN HE, Tsinghua University, Beijing, China

Sensing Resource Scheduling in 5G vRAN: An Elastic Approach

YULONG CHEN, Tsinghua University, Beijing, China
JUNCHEN GUO, Alibaba Group, Hangzhou, China
YIMIAO SUN, Tsinghua University, Beijing, China
HAIPENG YAO, Beijing University of Posts and Telecommunications, Beijing, China
YUNHAO LIU, Tsinghua University, Beijing, China
YUAN HE, Tsinghua University, Beijing, China

The emerging integrated sensing and communication (ISAC) technologies show great potential for 5G NR, offering a new wireless-sensing infrastructure paradigm. Users can benefit from pervasive sensing applications in various scenarios without communication penalties. Given the diverse demands for sensing resources across different sensing tasks, elastic resource scheduling becomes crucial, particularly when resources are constrained. However, existing approaches often treat users equally, limiting the applicability in dealing with diverse sensing tasks in the real world. In this article, we introduce ElaSe, a pioneering sensing technique that enables elastic and prompt scheduling of sensing resources. At the core of ElaSe is the exploration of the user's state to precisely determine the sensing resource requirements and schedule resources accordingly. We build the first model for matching sensing resources with sensing demands, and further propose a predictive scheduling scheme to eliminate delays by leveraging the 5G virtualized radio access network (vRAN). ElaSe has been implemented on a CPU-based 5G vRAN and commercial 5G user equipments. We conduct experiments to evaluate the performance of ElaSe under different settings. The results demonstrate that ElaSe outperforms the non-scheduling scheme, with a 34% reduction in trajectory tracking error and a 92% decrease in resource allocation error.

CCS Concepts: • **Networks** → **Programmable networks**; **Network management**; • **Hardware** → **Wireless integrated network sensors;**

Additional Key Words and Phrases: 5G, vRAN, wireless sensing, resource scheduling

ACM Reference Format:
Yulong Chen, Junchen Guo, Yimiao Sun, Haipeng Yao, Yunhao Liu, and Yuan He. 2025. Sensing Resource Scheduling in 5G vRAN: An Elastic Approach. *ACM Trans. Internet Things* 6, 3, Article 19 (July 2025), 24 pages.
https://doi.org/10.1145/3737886

This work is supported by the National Natural Science Foundation of China under grant No. 62425207 and No. U21B2007.
Authors' Contact Information: Yulong Chen, Tsinghua University, Beijing, China; e-mail: yl-chen22@mails.tsinghua.edu.cn; Junchen Guo, Alibaba Group, Hangzhou, Zhejiang, China; e-mail: juncguo@gmail.com; Yimiao Sun, Tsinghua University, Beijing, China; e-mail: sym21@mails.tsinghua.edu.cn; Haipeng Yao, Beijing University of Posts and Telecommunications, Beijing, China; e-mail: yaohaipeng@bupt.edu.cn; Yunhao Liu, Tsinghua University, Beijing, China; e-mail: yunhao@tsinghua.edu.cn; Yuan He (corresponding author), Tsinghua University, Beijing, China; e-mail: heyuan@tsinghua.edu.cn.

ACM 2577-6207/2025/07-ART19
https://doi.org/10.1145/3737886

1 Introduction

Due to the contact-less, privacy-protection, and even battery-free features, wireless sensing technologies have achieved great progress in the past few years. The mainstream **radio frequency**(**RF**) based wireless sensing has exploited wireless signals at various frequencies, ranging from LoRa [16, 43], RFID [22, 24], Wi-Fi [14, 31, 32, 48, 51–53] to mmWave [15, 49].

Although these works have demonstrated the feasibility of wireless-sensing applications, the method of deploying specific devices prohibits widespread popularization. The concept of ISAC proposed in mobile networks can solve the problem by building sensing system upon the pervasive base stations [19, 39]. Despite the pervasiveness, another useful feature of mobile networks is the centralized RF resource scheduling. Wi-Fi and backscatter networks [25, 26] widely adopt mechanisms like the contention-based CSMA/CA and the deep coupling between preamble signals and data packets. This will produce an unstable sampling rate of the sensing signals due to either the resource contention among sensing targets or the lack of valid data packets. Differently, the centralized resource scheduling and the separation of data and control channels in the mobile network can offer a stable sampling rate of the sensing signals with less multi-target contention and no data-transmission dependency.

In existing ISAC works with 4G LTE and 5G NR [10, 13, 23, 28, 29, 38, 42, 46, 47], we find that the majority of them have considered reducing the sensing errors by applying advanced signal processing algorithms [46, 47], emerging artificial intelligence technologies [10, 23], multi-stream infrastructure (e.g., multiple antennas [28] and multiple base stations [29]) when processing the LTE and NR **reference signals** (**RS**). However, lacking deep considerations of the intrinsic features of mobile networks, these topics are more or less discussed in other RF signal based works and similar solutions can be found.

Figure 1 shows a typical multi-target sensing scenario. The base station can schedule different amounts of sensing resources to terminals. We believe that the above-mentioned centralized resource scheduling is one of the intrinsic features to be well considered in ISAC. Take the existing works as an example, where fixed and averaged RS resources are shared among different UEs by default. Due to the limited total amount of available RS resources, one fast-moving UE will experience a sensing-quality degradation while there will be a waste of RS resources for the other static UEs. Since the RAN protocol stack integrates a data-plane scheduler that detects the data transmission demand of different UEs and schedules the limited data-plane resources among them under the principle of a long-term fairness, *can we design and integrate a similar RS scheduler to simultaneously boost the sensing quality and reduce the resource waste in the multi-target sensing scenarios?*

In traditional hardware-based black-box mobile networking infrastructure, only large device vendors can achieve this goal and efforts have to be spent on the hardware modification. Fortunately, the emerging 5G mobile networking technology named vRAN [11, 12] offers a more flexible way to achieve the interoperation for ISAC. First, vRAN adopts the idea of **Network Function Virtualization** (**NFV**) and implements the RAN protocol stack on general processors instead of dedicated chips in software. This gives us the opportunity of deploying a dedicated RS scheduler and upgrading it on demand. Besides, vRAN specifies a group of standardized interoperation interfaces, which enables the dynamic service loading and parameter adjustment. Plenty of works have leveraged the white-box 5G vRAN to enhance the data-transmission services like the cross-layer optimized video streaming [4, 6, 8, 44].

Though 5G vRAN shows promise in the wireless sensing applications, we have to tackle the following technical challenges before turning the above RS scheduler extension into a practical system:

— **First, how to match sensing demands with sensing resources?** Within the scope of our research, there is currently no work focusing on sensing resource scheduling in 5G networks.

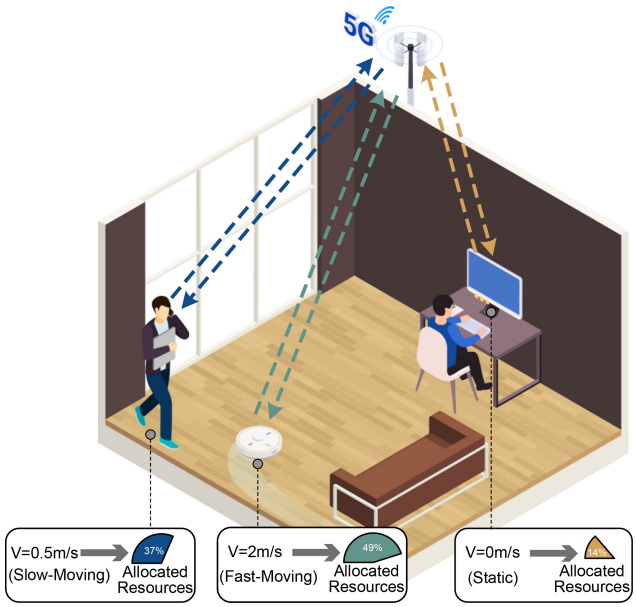


Fig. 1. Elastic sensing with ElaSe.

The criteria for estimating sensing resources, defining sensing demands, and matching them are not clear.

— **Second, how to eliminate scheduling delays?** While the scheduler adapts to changes in UE demands, scheduling delays remain unavoidable. The scheduling delays come from the execution time of the different algorithms and the requirement for recording some sampling points.

In this article, we propose ElaSe to address the above challenges. Our main contributions can be summarized as follows:

(1) We discuss the feasibility of the RS scheduling with 5G vRAN and provide our considerations about the scheduling space and principles (Section 2).
(2) We dissect the architecture of ElaSe and present its three main components named *UE state recognition*, *SRS resource scheduling*, and *Elimination of scheduling delays* (Section 3).
(3) We implement ElaSe with a 5G vRAN and commercial 5G UEs and conduct extensive evaluation experiments. The results show that ElaSe can elastically schedule appropriate sensing resources for UEs in different environments and motion states, thus achieving small tracking errors. The predictive scheduling scheme can further reduce allocation errors for sensing resources (Section 4).

Besides the above core contents, we also discuss the limitations of ElaSe and point out corresponding future directions in Section 5, and summarize the related works and their differences with ElaSe in Section 6. Finally, Section 7 presents a conclusion of our work.

2 Background

In this section, we first review the organization of the time-frequency domain resource in 5G, emphasizing the details of the resource for sensing (Section 2.1). Then, we introduce the background of 5G vRAN and discuss the feasibility of elastically scheduling the sensing resource in 5G

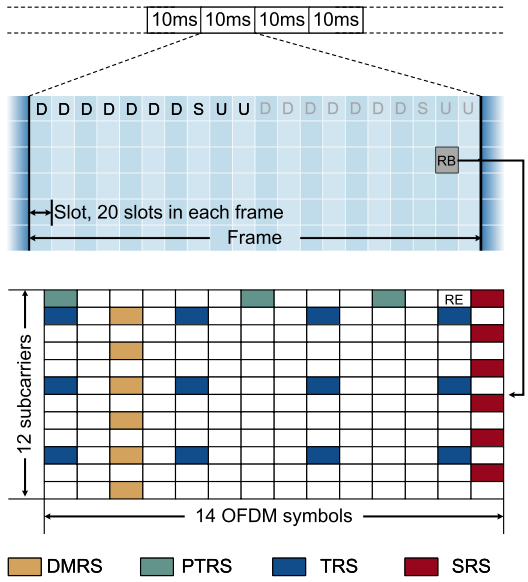


Fig. 2. 5G PHY frame structure and time-frequency domain patterns of four different RS in the U slot.

vRAN (Section 2.2). Finally, we highlight the design principles of our elastic resource scheduler (Section 2.3).

2.1 Organization of 5G Reference Signal

5G frame structure. 5G NR's physical layer (PHY) adopts the **Orthogonal Frequency Division Multiplexing** (**OFDM**) as the basic modulation scheme. As illustrated in Figure 2, the **resource elements** (**REs**) of 14 OFDM symbols in the time domain and 12 subcarriers in the frequency domain form the basic time-frequency unit named **Resource Block** (**RB**). The 5G slot is the time unit of one RB. Multiple continuous 5G slots are structured into 5G frames. Typically, one frame lasting 10 ms contains 20 slots with three different types: Downlink (D), Uplink (U), and Special (S). Figure 2 shows a basic slot format of "DDDDDDDSUU" in the commonly used **Time Division Duplex** (**TDD**) mode.

Different 5G reference signals. 5G NR adopts various types of RS for clock synchronization, frequency compensation and channel estimation for data transmission, including **Tracking Reference Signal** (**TRS**), **Phase Tracking Reference Signal** (**PTRS**), and **Demodulation Reference Signal** (**DMRS**), respectively. Moreover, for target sensing, it adopts the **Channel State Information Reference Signal** (**CSI-RS**) in the downlink channel and the **Sounding Reference Signal** (**SRS**) in the uplink channel. Our system utilizing 5G vRAN for elastic sensing only focuses on processing the continuous samples of SRS. Different RS have different time-frequency patterns regulated by the standards. We depict a classic case of these patterns in Figure 2, where SRS REs appear in the 14th OFDM symbol with a subcarrier spacing of 2 (termed as comb2 in 5G).

Extracting CSI from SRS. We can directly obtain the **Channel State Information** (**CSI**) for target sensing by concatenating the channel estimation results of SRS provided by the 5G PHY layer. The process of estimating $H(s, f)$, the element of the CSI matrix at the sth OFDM symbol and fth subcarrier, with the transmitted preamble $X(s, f)$ and received signal $Y(s, f)$ can be formulated as follows:

$$H(s, f) = \frac{Y(s, f)}{X(s, f)} = \sum_{l=1}^{L} a_l \, exp\left(-j2\pi \frac{d_l}{\lambda}\right) + n(s, f), \tag{1}$$

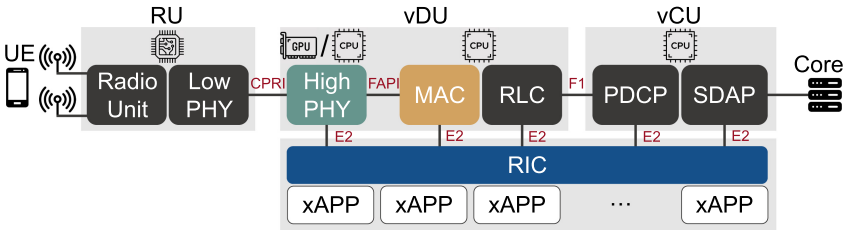


Fig. 3. Protocol stack of 5G vRAN: key modules run on general-purpose processors and interconnect with each other through standardized interfaces.

where a_l denotes the complex attenuation, d_l denotes the length of the lth signal path, L is the number of paths, λ is the wavelength of the signal, and $n(s, f)$ denotes noise.

2.2 Manipulating SRS for Sensing in 5G vRAN

Basic information of 5G vRAN. 5G vRAN is proposed to solve the closedness problem of the traditional RAN based on ASIC chips.

5G vRAN is usually used for deploying private 5G base stations. It is software-defined and designed to manage personal spaces, such as a factory or a private room, where sensing resources are limited. Scheduling sensing resources is necessary to improve the quality of service.

Figure 3 shows the typical data-plane structure of the 5G vRAN protocol stack, from bottom to top, including PHY layer, **Medium Access Control** (**MAC**) layer, **Radio Link Control** (**RLC**) layer, **Packet Data Convergence Protocol** (**PDCP**) layer, and **Service Data Adaptation Protocol** (**SDAP**) layer. Since RLC, PDCP, and SDAP are mainly responsible for the data transmission like packet retransmission, segmentation, security and quality-of-service enhancement, and so on, we here mainly review the details of PHY and MAC.

In 5G vRAN, PHY layer is separated into two parts: (i) the low PHY responsible for relatively fixed signal processing functions like FFT and iFFT; (ii) the high PHY responsible for upgradable modules like channel estimation, OFDM modulation, and channel coding. MAC layer centrally schedules physical resources for data transmission, RS, and broadcast signals. To integrate 5G vRAN with the capability of ISAC, we mainly focus on PHY's channel estimation module for the CSI retrieving and MAC's scheduler module for the elastic SRS scheduling.

Feasibility of on-demand SRS manipulating with RIC. The software-implemented protocol stack of 5G vRAN gives us the opportunity of manipulating SRS signals. Moreover, the O-RAN alliance has proposed the **RAN Intelligent Controller** (**RIC**) as the standardized interfaces for the vRAN interaction [2]. To obtain real-time information from vRAN and perform intelligent control, the customized xApps built with the RIC SDK interact with the RIC agent embedded in the vRAN via the E2 protocol. Although the current E2 protocol does not directly support the functionality of the SRS manipulation, we can easily conduct such extension based on its various implementations [30]. As depicted in Figure 4, with the help of RIC, an xApp designed for the ISAC application can deliver the SRS schedule to the MAC layer's RB scheduler and collect the estimated channel parameter to form the complete CSI matrix.

RIC can be classified into three categories based on its operating timescale: (i) Real-time RIC, with a timescale of less than 1 ms; (ii) Near-RT RIC, operating within 10 ms to 1 s; and (iii) Non-RT RIC, with a timescale greater than 1 s. RIC is capable of seamlessly accessing both application-layer and RAN-level data. Recent researches [9, 20] have also enabled the integration of AI/ML to enhance its services. The requirements for scheduling SRS and obtaining CSI are minimal, allowing these services to be seamlessly integrated into existing RIC frameworks. In our case, we only require the use of a Near-RT RIC.

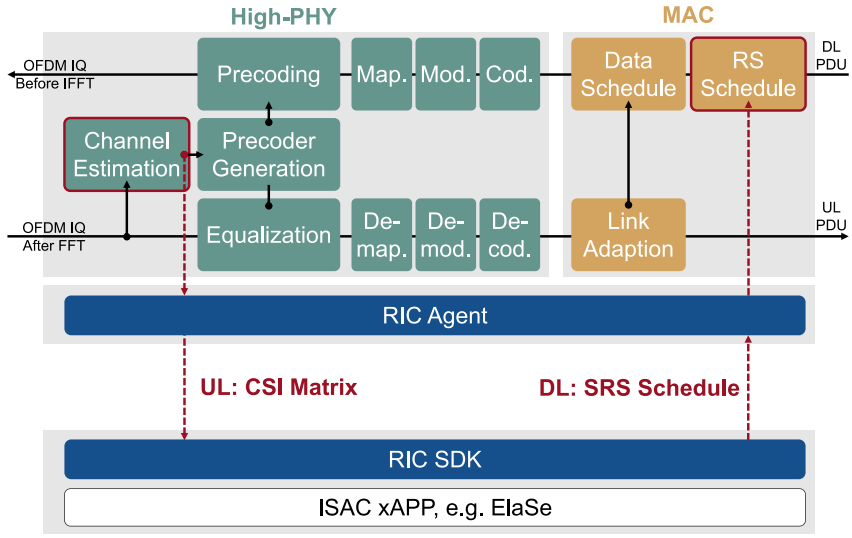


Fig. 4. Feasibility of manipulating 5G SRS: obtaining CSI from the channel estimation module in PHY and issuing elastic SRS allocation to the RS scheduler in MAC.

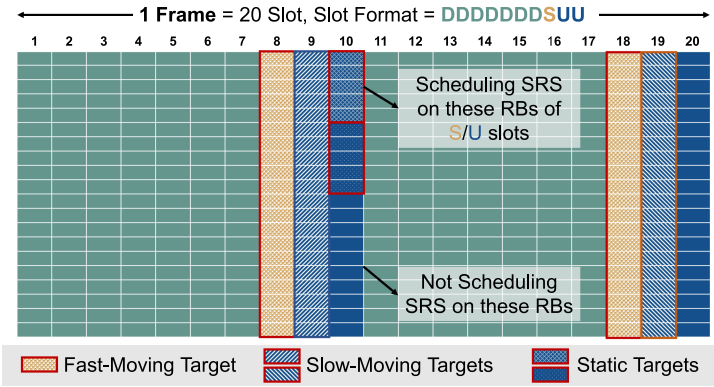


Fig. 5. Example of SRS scheduling for multiple targets: Faster target has more SRS resources in time domains.

2.3 Principles of Elastic Sensing Resource Scheduling

Before digging into the detailed designs of ElaSe, we introduce the definition of "Elastic Scheduling" and its criteria. We define **"Elastic Scheduling"** as follows. Elastic scheduling refers to the adaptive and real-time configuration of sensing resources aimed at minimizing resource waste and maximizing service quality. This is achieved by monitoring target demands and leveraging prediction algorithms. The process involves the following two criteria:

Dynamic Resource Scaling: Our research focuses on adjusting resource allocation for users to minimize the number of SRS slots. The scaling of resources is based on the real demand. If the demands for resources exceed the total capacity, algorithms can calculate weights to schedule sensing resources among users rationally.

Real-Time Availability: The base station monitors the sensing demands of users. There is a delay between changes in user sensing demands and resource adjustments. This delay includes the

sampling time for monitoring and the scheduling activation time. The scheduling activation time is constrained by the performance of the RIC, which is typically within 10 ms and can be ignored. The system should eliminate the major delay from sampling.

Then we introduce our considerations in the SRS scheduling pattern, range, and logic. For a typical setting of the "DDDDDDDSUU" slot format, SRS can be scheduled in the S and U slots. For the standard 100 MHz bandwidth in 5G, each slot contains 273 RBs in the frequency domain. To provide a more effective SRS scheduling, we adopt the following principles.

Scheduling Pattern: First, for the SRS comb pattern, we choose the default *comb2* instead of other available options such as *comb4*. Second, for the SRS symbol pattern, we adopt at most one SRS symbol in each slot. The rationale behind our selections is the tradeoff between the gain and cost: obtaining a denser CSI sample in one slot yields a limited sensing gain. Third, we choose continuous RB allocation each schedule instead of the dispersed one, i.e., allocating [*RB_start*, *RB_num*] for each user at each slot. According to the 5G NR standard, the value of *RB_start* can be arbitrary while the value of *RB_num* must be chosen from the non-contiguous values in a predefined table (Table 6.4.1.4.3-1 of 3GPP TS38.211 [1]). It's worth noting that the maximum value of *RB_num* is 272 instead of 273 because the number needs to be a multiple of 4.

Scheduling Range: Here we mainly discuss the sparsest schedule and the densest schedule of SRS RBs in the time-frequency domain. For the sparsest one, we comply with the reference implementation of a most popular vRAN system named OAI. In OAI, the SRS is scheduled once at the end of each frame for channel state estimation, with a minimum bandwidth of four RBs. This sparsest schedule can produce CSI samples with a sampling rate of 100 Hz and a sampling bandwidth of 1.5 MHz. For the most densest one, we can allocate the entire SRS to one UE. This densest schedule can produce CSI samples with a sampling rate of 600 Hz and a sampling bandwidth of 100 MHz.

Scheduling Logic: Since the SRS resources are limited, we expect to elastically schedule them among multiple targets. Our basic scheduling logic is to allocate more resources to the targets that need these resources more, e.g., fast-moving targets. For slow-moving targets or static targets, we reduce the resources scheduled for them. When the total number of SRS RBs required by the sensing targets of the entire system is fewer than the number of all available SRS RBs, we free up the remaining RBs for pure communication usage.

3 System Design

3.1 ElaSe Overview

The overview of ElaSe is shown in Figure 6. ElaSe implements a system for identifying UE's state and scheduling SRS sensing resources. We introduce each module below:

- **UE state recognition.** ElaSe aims to accurately identify the dynamic demands of UEs of sensing resources. To achieve this objective, we leverage the UE's velocity as a state indicator, recognizing that different velocities reflect different states and require varying SRS resources. To recognize the UE's velocity, we utilize the existing low sampling rate SRS in communication and derive the velocity information from the CSI. It should be emphasized that the velocity estimation algorithm requires a certain number of sampling points, which necessitates a certain amount of time to gather. To address this, we implement a periodic strategy and find out the appropriate cycle length.
- **SRS resource scheduling.** After recognizing the UE's state, it's essential to establish a mapping relationship between the state and the sensing resources and allocate resources effectively for each target. Each UE can request a specific SRS bandwidth based on its task requirements or use the default configuration. We achieve dynamic adaptation by mapping

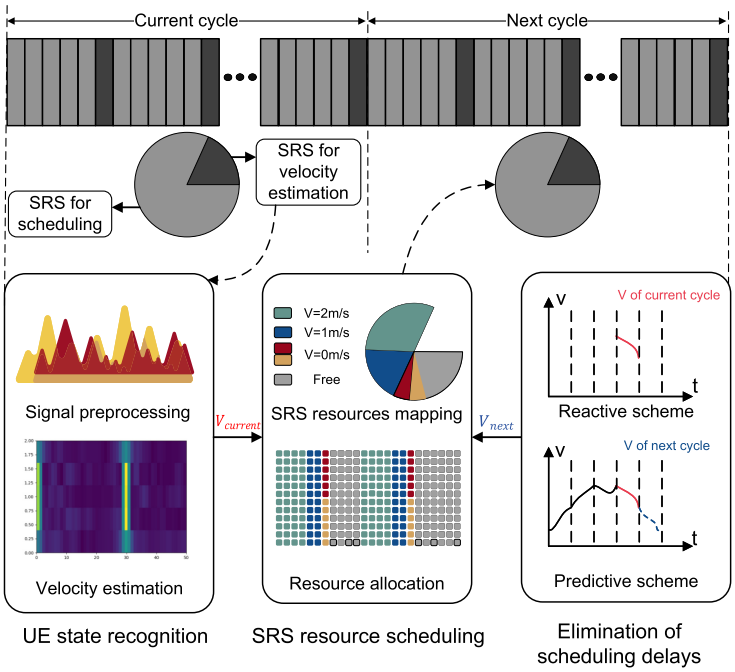


Fig. 6. Design overview of ElaSe.

the UE's velocity to the sampling rate of SRS. The base station aims to minimize the number of time slots used in scheduling sensing resources. Additionally, in scenarios of sensing resource oversubscription, we propose a weight-based allocation algorithm specifically tailored for sensing tasks to optimize resource utilization efficiency.

— **Elimination of scheduling delays.** The velocity of each UE is calculated by the base station at the end of each cycle. One intuitive approach is to employ an immediate response scheduling strategy, where the current velocity is utilized to allocate sensing resources for the next cycle. We call it Reactive Scheduling Scheme. However, this scheme introduces a significant delay between the UE acquiring sensing resources and its state change. To eliminate delays, we perform a delay analysis and propose a Predictive Scheduling Scheme.

— **Multi-target sensing.** Finally, ElaSe accomplishes different sensing tasks using the different amounts of SRS resources scheduled to different targets. In this context, we consider localization and tracking to be multi-target sensing tasks.

3.2 UE State Recognition

In Section 2.3, we mentioned that in the original 5G communication system, each frame assigns one SRS for channel quality estimation. This SRS has a sampling rate of 100 Hz. We utilize this periodic SRS to recognize the UE's state. We introduce the velocity estimation algorithm in this section.

3.2.1 Velocity Estimation Algorithm. Prior researches [17, 27, 41] have proposed methods for estimating the velocity of device movement using CSI exclusively. This method proposed by C^2IL [17] is based on the electromagnetic standing wave field. C^2IL finds through electromagnetic wave propagation theory and measurements that when an antenna traverses an indoor space at a velocity of v, a periodically ripple-like pattern with a frequency of f_0 emerges, following the relationship

$f_0 = \frac{2*v}{\lambda}$, where λ represents the wavelength of the electromagnetic wave. C^2IL utilizes a CSI matrix over time on one spatial stream and one subcarrier to extract the frequency f after undergoing data preprocessing, noise cancellation, fading enhancement, and frequency estimation. By extracting f from multiple subcarriers and selecting the median value f_0 as the final result. Then the moving velocity is estimated by

$$v = \frac{\lambda * f_0}{2} \tag{2}$$

3.2.2 Periodic Strategy. The velocity estimation algorithm requires a certain number of sampling points. However, adopting a pipeline strategy would introduce more computational overhead. Considering the need for resource scheduling, we adopt a periodic strategy. At the end of each cycle, the base station calculates the velocity of the UE in the current cycle and schedules the SRS resources for the next cycle. If the cycle is set too short, there would not be enough CSI samples to identify the UE's state, and the scheduling space for the next cycle would be insufficient. On the other hand, if the cycle is set too long, the estimated average velocity of the UE in the current cycle could not represent the UE's state accurately, leading to inadequate timely performance of the system. We will measure the impact of different cycle lengths in our experiments.

3.3 SRS Resource Scheduling

3.3.1 Mapping Velocity to SRS Resources. After obtaining the velocity of each UE, it is necessary to convert the velocity into sensing resource demand. Regarding the bandwidth of SRS, each UE can transmit its required bandwidth, ranging from 0 to 100 MHz, to the base station. ElaSe will record the identifier of the UE and the bandwidth it requests. Different bandwidths can be adjusted to suit different sensing tasks. In the case where a UE does not make an application, by default, we allocate half bandwidth to stationary UEs and all bandwidth to moving UEs. Then, let's delve into the sampling rate. According to [17], the typical velocity of walking indoors ranges from 0.8 to 1.6 m/s, with a maximum value of 2 m/s. The velocity value represents the displacement of the UE per unit time, while the sampling rate represents the number of samples taken within the same unit of time. Assuming the UE is sampled once after the same displacement, we establish a positive correlation between the sampling rate and velocity value. Let's denote the sampling rate as f_s, the scaling coefficient as k, and the velocity value as v. The relationship among them can be expressed as follows:

$$fs = k * v \tag{3}$$

To determine the coefficient k, we conduct a set of pre-experiments to measure the relationship between tracking error and sampling rate. We set the velocities of UE from 0 m/s to 2 m/s, and the sampling rate from 20 Hz to 200 Hz. We use the sampling method when the rate is less than the sparsest rate and keep the bandwidth of SRS 100 MHz. The localization method adopts a modified super-resolution method from SpotFi [21].

The experimental results are shown in Figure 7. Based on the experimental findings, we determine the value of the scaling coefficient as k=100 to keep the average trajectory tracking error less than 70 cm.

In infrequent circumstances, the velocity of indoor user equipment might exceed 2 m/s. Continuously increasing the sampling rate linearly will result in a relatively small gain in the sensing effect. As shown in Figure 5, the SRS is distributed across three consecutive slots within one subframe. Hence, when the sampling rate is higher than 200 Hz, a UE will be repeatedly sampled in a short time interval, thereby leading to a smaller gain in the sensing effect. We employ a logarithmic function to fit this situation and ensure smooth transitions at the segmentation points:

$$fs = 200 * \ln(v) + 61, v > 2m/s \tag{4}$$

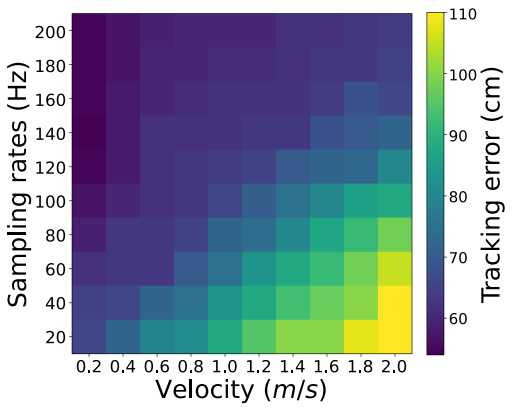


Fig. 7. Variant tracking errors under different velocities and sampling rates.

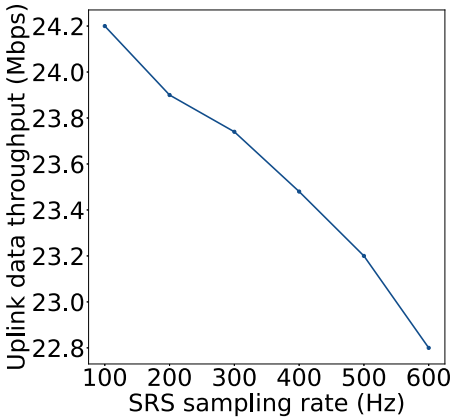


Fig. 8. Uplink data throughput using SRS with different time slot numbers.

Additionally, for UEs with extremely low velocities, it is essential to maintain a minimum SRS sampling rate. Therefore, we have set a threshold of 50 for the minimum sampling rate. To summarize, the sampling rate is given by

$$fs = \begin{cases} 50 & , v \leq 0.5m/s \\ 100 * v & , 0.5m/s < v \leq 2m/s \\ 200 * \ln(v) + 61 & , v > 2m/s \end{cases} \tag{5}$$

3.3.2 Resource Allocation Algorithm. Once the demands of SRS resources for the next cycle are determined, the base station considers scheduling which SRS resources in the time and frequency dimensions for each UE. This resource allocation problem can be transformed into a two-dimensional coloring problem. In the time domain, there are 500 time slots per second available for SRS resources, while in the frequency domain, a bandwidth of 272 RBs is allocated. It is possible to divide and allocate bandwidth to different UEs within the same time slot, but the allocated bandwidth for a single UE needs to be continuous.

Effects of SRS slot numbers on data throughput. We conduct experiments to measure uplink data throughput using SRS with varying time slots. Figure 8 illustrates that the uplink data throughput decreases as SRS slot numbers increase. The throughput with 600 SRS slots shows a 5.8% reduction in comparison with that with 100 slots. This is due to the fact that in the OFDM structure, sensing resources and communication resources are mutually exclusive.

Therefore, to minimize the total number of SRS time slots, the resource allocation algorithm combines the sensing resource demands of all UEs. If the combined SRS bandwidth required by multiple UEs does not exceed 100 MHz, they can be merged into a single time slot. In our resource mapping scheme, two stationary UEs can each occupy 50 MHz bandwidth within the same time slot. It is important to note that different resource mapping schemes exist, allowing for a finer distinction of UE SRS bandwidth requirements. In such cases, multiple UE SRSs with a total bandwidth not exceeding the upper limit can be allocated in the same time slot.

When multiple UEs require a large amount of resources, there may be a challenge of overall demand exceeding the available resource limit. In this case, traditional communication or computing resource scheduling scenarios typically handle best-effort applications that focus primarily on throughput and do not have strict delay constraints. Therefore, schedulers often use proportional

fair algorithms to reduce resources in the same proportion to meet overall demand. However, in sensing resource scheduling, sensing tasks usually have timeliness requirements, so resource allocation algorithms should be able to provide a QoS guarantee.

Giving the same weight to all targets is not ideal in sensing resource scheduling. Our insight is that during periods of changes in motion state, UEs have a high priority for resource allocation and require prompt processing by the scheduler. In particular, UEs with increasing resource demands should be assigned greater weights. We use the sampling rate as an indicator to reflect changes in resource demand and assign higher weights to UEs with increasing sampling rates.

Additionally, to address the problem of starvation, we introduce a historical deficit compensation mechanism when calculating weights. The weight calculation for each target is

$$w_i = 1 + \frac{max(fs_T - fs_{T-1}, 0)}{100} + \frac{fs_{(demand,T-1)}}{fs_{(allocate,T-1)}} \tag{6}$$

In the weight calculation, the second term involves subtracting the previous cycle's sampling rate from the current cycle's sampling rate, thereby increasing the weight for UEs with rising sampling rates. The third term represents the ratio of resource demand to allocation in the previous cycle, which serves to compensate UEs with low resource satisfaction rates.

3.4 Elimination of Scheduling Delays

3.4.1 Reactive Scheduling Scheme. The Reactive Scheduling Scheme functions effectively when the UE's velocity remains constant. However, when the UE experiences changes in velocity, such as acceleration, the velocity in the next cycle exceeds the estimated velocity in the current cycle. Consequently, the sensing resources allocated based on the current cycle's velocity may prove insufficient to meet the actual demand. This scheme introduces a delay of one cycle between the change in the UE's velocity and the adjustment of allocated sensing resources. When considering resource allocation as a service to meet evolving demands over time, such delays can pose challenges for subsequent sensing applications.

The reason behind the one-cycle delay in the Reactive Scheduling Scheme is that the base station requires one cycle to obtain CSI for determining the UE's velocity. Therefore, when scheduling sensing resources for the UE in the next cycle, the UE's velocity in that cycle remains unknown. In response to this challenge, we propose the Predictive Scheduling Scheme.

3.4.2 Impact of Scheduling Delays on Real-Time Applications. In practical real-time applications, different levels of latency have different impacts on system performance and user experience. Focusing on the human localization and tracking application in ElaSe, we discuss the specific effects of different latency levels as follows:

Millisecond-Level Latency (less than 100ms). For human motion, even at the highest speeds (typically less than 8m/s in general environments), latency under 100ms is nearly negligible and does not significantly affect the system's sensing accuracy. However, in high-speed scenarios, such as autonomous driving, applications require much stricter real-time responses. A 100ms latency in these scenarios could prevent the system from accurately recognizing dynamic environmental changes, potentially causing safety hazards.

Second-Level Latency (100ms - 1s). On a time scale of seconds, humans can change their motion states or trajectories, making second-level latency a major factor that undermines tracking accuracy. Delays within this range can introduce big errors in tracking results, degrading the performance of the sensing system. In addition, ElaSe has a sampling period between 500ms and 1s, resulting in scheduling delays within this range. Therefore, eliminating second-level latency is the target for the predictive scheduling scheme.

Table 1. Comparison of Prediction Algorithms

Algorithm	Noise Robustness	Computational Complexity	Dynamic Modeling Capability
Moving Average	Low	$O(1)$	None
Kalman Filter	High	$O(d^3)$ (d: number of states)	Strong
Particle Filter	High	$O(N_p)$ (N_p: number of particles)	Strong

Higher Levels of Latency (1s - 10s). When latency exceeds 1 second, it severely destroys the real-time capabilities of the system. On one hand, the base station will be unable to keep pace with the dynamic changes in human motion, leading to the failure of the sensing results. On the other hand, users will perceive lag in the tracking service, lose confidence in the system's performance, and may even choose to discontinue using the service entirely.

3.4.3 Predictive Scheduling Scheme. Although the base station lacks knowledge of the UE's velocity in the next cycle during resource scheduling, it can gather historical velocity data of the UE in multiple past cycles. By utilizing this historical data, the base station can compute the current acceleration of the UE and use it to estimate the UE's velocity in the next cycle. Furthermore, to enhance the accuracy of UE velocity prediction, we employ a Kalman filter that combines prior information with measurement information. This integrated approach enables more precise state estimation. The velocity prediction algorithm leverages the acceleration values and Kalman filtering to predict the velocity in the next cycle. At the end of the next cycle, the velocity estimation results are fed back into the Kalman filter. This iterative process aims to continuously optimize the prediction algorithm and enhance its accuracy in predicting. By incorporating this feedback mechanism, the prediction algorithm can rectify and improve its predictions based on real observed velocity data.

Advantages of choosing Kalman filter. Prediction scheduling can utilize the moving average algorithm, particle filter algorithm, or Kalman filter algorithm. Table 1 illustrates their characteristics. The moving average algorithm is simple but responds slowly to changes in user states and cannot handle dynamically changing systems. The particle filter algorithm is capable of handling complex dynamic scenarios. However, as the number of particles is typically in the thousands, it causes significant computational overhead, making it unsuitable for real-time prediction tasks.

In our velocity prediction scenarios, states in the Kalman filter can be modeled as [speed, acceleration], resulting in fewer states and lower computational overhead. We choose the Kalman filter as the state prediction algorithm because of its robust dynamic modeling capabilities, low computational complexity, and strong noise robustness. Given the characteristics of the prediction scenarios and real-time requirements, the Kalman filter algorithm is more suitable for prediction scheduling.

Theoretical analysis of scheduling delay. Table 2 illustrates the difference between the Reactive Scheduling Scheme and the Predictive Scheduling Scheme. In the scenario described, a UE remains stationary during the first cycle, accelerates from the 2nd to the 3rd cycle, and moves rapidly from the 4th to the 6th cycle. As the UE's velocity is high in the 4th cycle, it should ideally obtain a high sampling rate SRS during that cycle. However, due to the inherent delay of the Reactive Scheduling Scheme, the base station only recognizes the UE's velocity is high at the end of the 4th cycle, causing the UE to obtain the high sampling rate SRS in the 5th cycle. In contrast, the Predictive Scheduling Scheme predicts the velocity of next cycle at the end of the 3rd cycle, thereby eliminating scheduling delays.

Table 2. A Scenario where UE Accelerates First and then Moves Uniformly

Cycle	1	2	3	4	5	6
UE's acceleration	Zero	Positive	Positive	Zero	Zero	Zero
UE's velocity	Zero	Low	Low	High	High	High
Reactive scheme	\	Recognize velocity in 2	Recognize velocity in 3	Recognize velocity in 4	Schedule a high sampling rate SRS	\
Predictive scheme	\	Predict velocity in 3	Predict velocity in 4	Schedule a high sampling rate SRS	\	\

Predictive Scheduling Scheme eliminates the scheduling delay.

3.5 Multi-Target Sensing: Localizing and Tracking

After the resource scheduling process, the base station extracts CSI from the received SRS, analyzing it to determine the UE's serial number, time slot, and associated frequency. Leveraging this information, the base station can offer customized sensing applications for individual users.

In ElaSe, we take localizing and tracking as sensing tasks. We adopt a super-resolution localization algorithm inspired by SpotFi.

This algorithm is specifically designed for scenarios where there are few antennas and many subcarriers, aligning with our system's characteristics. SpotFi introduces a virtual sensor array that combines the CSI values from each subcarrier and antenna. By leveraging this virtual array, it jointly estimates the **Angle of Arrival** (**AoA**) and **Time of Flight** (**ToF**) for each multipath signal. Then it finds out the AoA of the direct path. Additionally, SpotFi utilizes the **Received Signal Strength Indicator** (**RSSI**) from the **access point** (**AP**) to estimate the distance between the UE and the AP. Considering the unique features of the 5G vRAN system, we have made two specific modifications:

(1) Although the RSSI value of the uplink signal can be obtained at the 5G vRAN base station, we conducted measurements and discovered that the CSI amplitude also contains distance information. In the same room, there exists a consistent relationship between the CSI amplitude and the distance from the UE to the base station. Therefore, we utilize the CSI amplitude to estimate the distance between the UE and the base station.
(2) The SpotFi algorithm can achieve localization using a single AP and one CSI sampling. To reduce localization errors, SpotFi incorporates the results of three APs from one sampling. Differently, ElaSe comprises only one base station, but it has positioning results from high-frequency sampling. Considering the limited velocity of human movement indoors, we employ the continuity of movement to discard outliers and mitigate localization errors.

4 Implementation and Evaluation

4.1 Implementation

First, we build a complete 5G testbed including a 5G base station and multiple commercial 5G UEs. The 5G base station consists of the 5G vRAN and the 5G core network. Details of these components are listed as follows:

— **5G vRAN:** We adopt the most widely deployed open-sourced vRAN project named **OpenAirInterface** (**OAI**). The hardware-based RU of OAI runs a USRP N310 with 2 TX antennas and 2 RX antennas. The software-based vDU and vCU of OAI run on a PC equipped with an i9-13900K CPU and 16GB RAM. The USRP and the PC are connected through a 10 GBE optic fiber due to the heavy traffic of the raw IQ samples. The vRAN test is set up with a 100 MHz bandwidth at 3.9 GHz (N77 band) with a 30 KHz sub-carrier spacing.

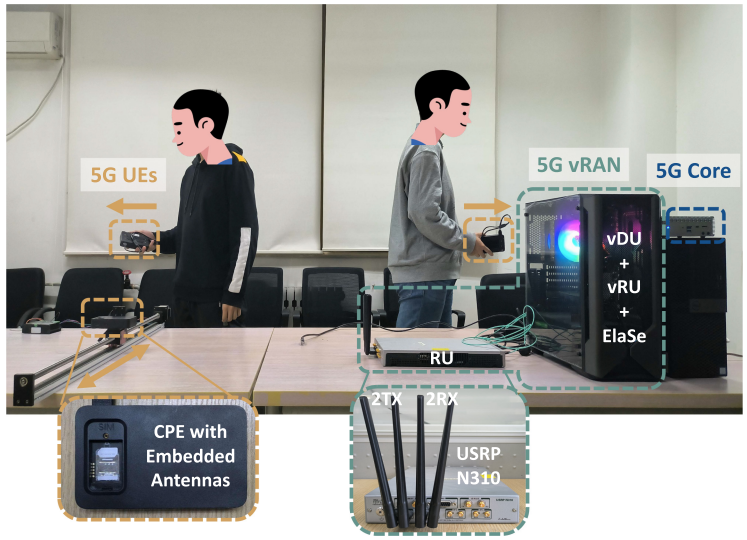


Fig. 9. Experiment scenario.

— **5G core network:** We deploy the commercial version of free5GC for its better stability and easier operation than the open-sourced one. Since ElaSe requires the raw CSI samples for a better sensing quality, we do not use the **Location Management Function** (**LMF**) provided by the 5G core. The core network runs on a Linux industrial computer directly connected to the vRAN PC with a 2.5 GBE network cable.
— **5G UEs:** We choose the commercial 5G **Customer Premise Equipment** (**CPE**) instead of the smartphone because it can access the private 5G network in non-public network frequency bands like N77. No hardware and software modifications are applied to these UEs in ElaSe.

Second, we implement the three key components of ElaSe and deploy them on the same PC that hosts the OAI's protocol stack.

Figure 9 shows the real-world testbed deployment of ElaSe. All the vRAN hardware is statically placed on a wooden table inside a conference room, including the four antennas of the USRP. The moving UEs are either held by volunteers or carried on a guide rail. The volunteers randomly walk around the room with different velocities. The guide rail can precisely control the motion of the UE with specific acceleration or velocity, thereby obtaining accurate ground truth. All the experiments are IRB-approved, and all data are anonymized.

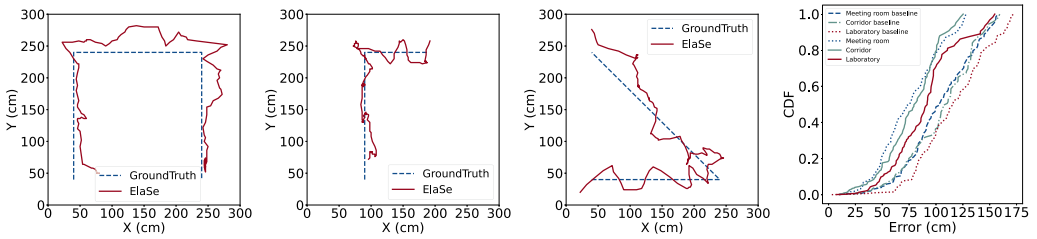
4.2 Baselines and Comparison Methodologies

The baseline is a scheme that uses the original SRS with a sampling rate of 100 Hz and a whole bandwidth, evenly allocated to UEs. We compare the baseline with two other scheduling schemes: reactive scheme and predictive scheme. We evaluate the weight-based allocation algorithm in the scenario of resource oversubscription.

We use trajectory tracking error and resource allocation error as the metrics for evaluation. The trajectory tracking error measures the end-to-end performance of ElaSe. The resource allocation error is defined as the absolute difference between the number of SRS allocated to one UE and the real demand in one cycle. The number of SRS is derived by multiplying the number of RBs by the number of slots. We use resource allocation error as the metric because ElaSe focuses on elastic

Table 3. Experimental Conditions and Parameters

Attribute	Meeting Room	Corridor	Laboratory
Room Size	5m × 4.2m	5m × 2m	5m × 4m
Multipath Effect	Some multipath	Few multipath	Rich multipath
Signal Interference	WiFi signals exist in 2.4G and 5G bands. No signal in the N77 band (3.3GHz–4.2GHz)		
Number of UEs	3		
Test Time	30 s		
Hardware Specifications	The base station is equipped with an i9-13900K CPU, 16 GB RAM and no GPU. The operating system is Ubuntu 22.04. The RU is USRP N310 with 2TX and 2RX.		



(a) Meeting room (b) Corridor (c) Laboratory (d) TE analysis

Fig. 10. Overall performance of trajectory tracking error in different environments.

and prompt sensing resource scheduling. Therefore, if the delay in scheduling increases, the value of allocation error will be larger.

4.3 Overall Performance

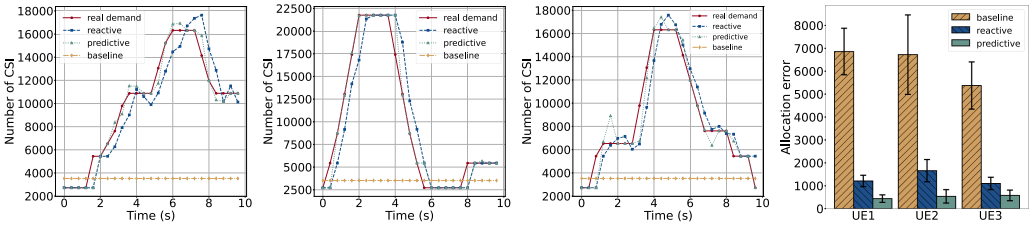
Trajectory tracking error. Table 3 describes the experimental conditions and parameters. We first evaluate ElaSe's performance of trajectory tracking in various environments, as illustrated in Figure 10. During each experiment, we adopt the predictive scheduling scheme.

Figure 10(a)–(c) depicts the system's tracking trajectory of UEs in these three environments, compared with the ground truth. Additionally, Figure 10(d) presents the **cumulative distribution function** (**CDF**) of the tracking error. It is evident that the laboratory environment showcases the highest tracking error, as indicated in Figure 10(c), mainly attributed to the heavy multipath. In the meeting room, corridor, and laboratory, the median trajectory tracking error is 60.24 cm, 63.57 cm, and 78.31 cm, respectively.

In the baseline scheme, the median trajectory tracking error is 90.25 cm, 94.53 cm, and 110.69 cm, respectively. The median trajectory tracking error of ElaSe in the predictive scheme is 34% lower than that of the baseline scheme.

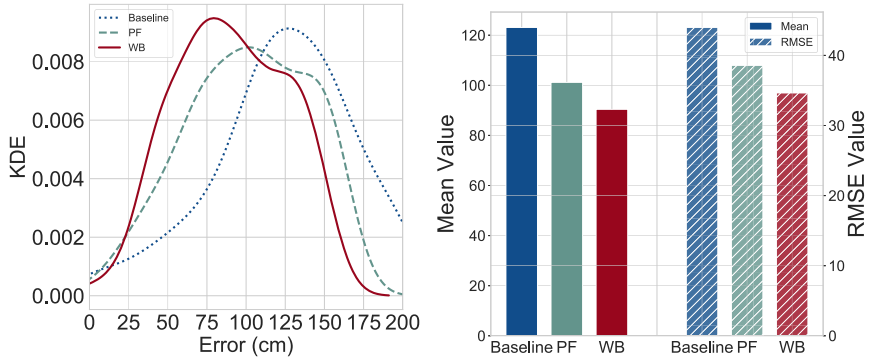
Resource allocation error. Then we evaluate the performance of our allocation scheme under different motion states of UEs. The acceleration and velocity of UEs vary in different motion states. We analyze ElaSe's performance of resource allocation in a 10-second motion of each UE.

Figure 11(a)–(c) illustrates the real demand for SRS by the three UEs and the changes in allocation over time in reactive, predictive, and baseline schemes. By comparing the real demand with the reactive scheme, we observe that the allocation results of the reactive scheme are generally close to the real demand in the previous cycle. This indicates a scheduling delay in the reactive scheme. In Figure 11(a), when the UE's velocity changes most of the time, it is evident that the allocation curve of the reactive scheme lags behind the real demand. However, in Figure 11(b), when the UE's velocity remains unchanged, the results of the reactive scheme align with the real demand value.



(a) UE 1 (b) UE 2 (c) UE 3 (d) AE statistic

Fig. 11. Overall performance of resource allocation error in different motion states.



(a) Kernel density estimation of tracking errors (b) MEAN and RMSE of tracking errors

Fig. 12. Trajectory tracking error using different resource allocation algorithms.

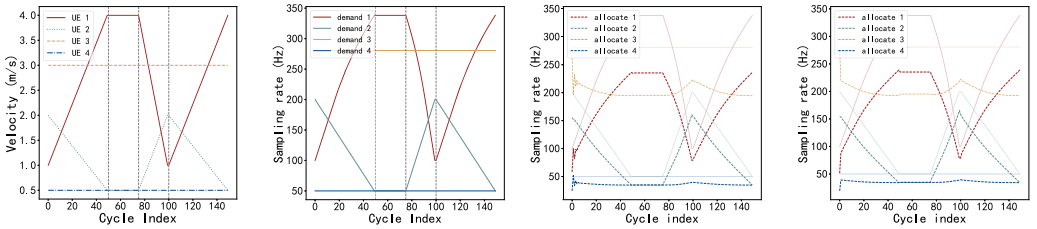
Figure 11(d) provides statistical results of the resource allocation error of the three schemes. The resource allocation error of the reactive scheme is 78% lower than that of baseline scheme, furthermore, the resource allocation error of the predictive scheme is 92% lower than that of the baseline scheme. This is because the baseline scheme does not consider the state changes of the UE, resulting in insufficient sensing resources being allocated to the UE. The results illustrate that the prediction algorithm effectively predicts the velocity of the UE in the next cycle.

4.4 Performance of Weight-Based Allocation Algorithm

In scenarios where resources are oversubscribed, ElaSe uses the weight-based allocation algorithm. We compare it with the proportional fair algorithm and baseline scheme with no scheduling.

4.4.1 Performance of Error Control Capability. To create a scenario of resource oversubscription, we simulate two UE accesses in the SRS scheduler of vRAN and maintain a speed of 1m/s. We connect three real UEs to the base station and record the trajectory tracking error as the real UEs move. The resource allocation algorithm used for simulated and real UEs is identical.

Figure 12(a) presents the kernel density estimation of tracking errors under different resource allocation algorithms, revealing the probability distribution characteristics of the tracking errors. The baseline algorithm exhibits the poorest performance, with more than half of the errors exceeding 120 cm. In comparison, the proportional fair algorithm shows relatively concentrated errors, with 38% of the errors falling between 100 and 150 cm. The weight-based allocation algorithm delivers the best performance, with errors primarily concentrated around 80 cm.



(a) velocity of four UEs (b) Sampling rate corresponding to velocity (c) Demand value and allocation value of Weight based algorithm (d) Demand value and allocation value of PF algorithm

Fig. 13. Evaluation of long-term fairness.

Table 4. Comparison of Resource Satisfaction Rates: WB vs. PF Allocation Algorithm

User Equipment	Weight-Based Allocation Algorithm			Proportional Fair Allocation Algorithm		
	Max	Min	Average	Max	Min	Average
1	0.9390	0.5821	0.7240	0.8287	0.5136	0.7332
2	0.8023	0.6928	0.7203	0.8259	0.6854	0.7193
3	0.9340	0.6928	0.7191	0.9799	0.6854	0.7135
4	1.0000	0.4848	0.7177	0.7927	0.3845	0.7095

According to Figure 12(b), the average tracking errors for the baseline, proportional fair, and weight-based allocation algorithms are 123.09 cm, 101.16 cm, and 90.39 cm, respectively. Meanwhile, the root mean square errors for these algorithms are 43.98, 38.58, and 34.65, respectively. The weight-based allocation algorithm reduces the density of large errors by effectively allocating sensing resources, making it more suitable for tasks related to sensing resource allocation.

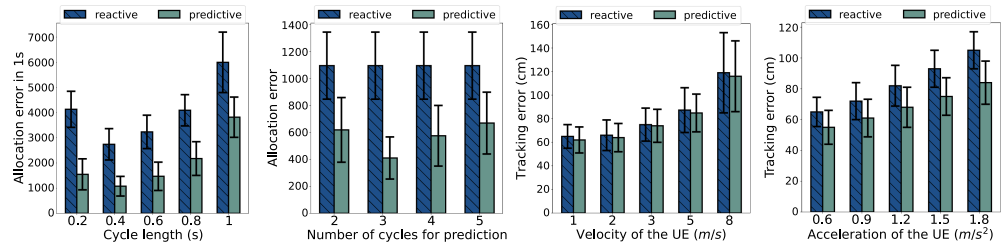
4.4.2 Performance of Long-Term Fairness. As shown in Equation (6), when calculating the weight, the third term of the equation is the resource satisfaction rate of the UE. We define resource satisfaction rate of the ith user in the Tth cycle $R_{i,T} = \frac{fs_{(allocate,T)}}{fs_{(demand,T)}}, 0 < R \leq 1$. To evaluate the ability of the weight-based allocation algorithm to maintain fairness, we simulate the dynamic behavior of four UEs and record their resource allocation. The simulation time is set to 150 cycles to evaluate the long-term performance.

Figure 13(a) illustrates the velocity variations of the four UEs. The time is divided into four phases: Cycles 0 to 50, 50 to 75, 75 to 100, and 100 to 150. Under this scenario, UE 1 and 3 are considered high-speed devices that require more sensing resources.

Figure 13(b) shows the sampling rate of the UEs calculated by the velocity, which can be called demand value. Based on this, we compute the actual allocated sampling rates using the weight-based allocation algorithm and the proportional fair allocation algorithm. As shown in Figure 13(c)(d), due to resource over-subscription, the allocated value to each UE is lower than the demand value.

Table 4 shows the resource satisfaction rate statistics of two allocation algorithms. It can be seen that the resource satisfaction rate of the weight-based allocation algorithm is higher than that of the proportional fair algorithm in terms of maximum, minimum, and average values. This demonstrates the former's ability to ensure fairness.

For the minimum satisfaction rate, the weight-based allocation algorithm maintains generally higher levels. For example, the minimum value for UE 4 is 0.4848 with the weight-based allocation, while the minimum value with the proportional fair algorithm for the same equipment is only



(a) Length of scheduling cycle (b) Length of prediction window (c) UE velocity (d) UE acceleration

Fig. 14. Evaluation of the four main factors' impacts on the performance of ElaSe.

0.3845. This indicates that the former one is better at ensuring a certain level of resource satisfaction under resource scarcity. In conclusion, the weight-based allocation algorithm is effective in maintaining long-term fairness.

4.5 Impacting Factors

In order to evaluate the system's performance comprehensively, we have identified some internal and external factors that affect it. Internal factors include the length of the scheduling cycle and the length of the prediction window, which affect the accuracy of UE state recognition and velocity prediction. External factors include the velocity and acceleration of UE motion, which impact trajectory tracking.

Length of the scheduling cycle. We set different cycle lengths while keeping the UE's motion state the same. We normalize the resource allocation error in one cycle to obtain the allocation error in one second. The experimental results are presented in Figure 14(a).

For the reactive scheduling scheme, the resource allocation error initially decreases and then increases as the cycle length grows, reaching its minimum value at a cycle length of 0.4 seconds. When the cycle length is short, there are insufficient sampling points for state recognition, leading to inaccurate velocity calculation. Besides, when the cycle spans a longer duration, the recognition result is the average velocity during past period, which causes resource allocation error because the UE may change state in one long cycle.

Length of the prediction window. We evaluate the impact of the length of the prediction window on allocation error. A smaller resource allocation error indicates more accurate velocity prediction. We set the cycle length to 0.4 seconds and vary the length of the prediction window from 2 cycles to 5 cycles in our experiments. The results are shown in Figure 14(b).

The resource allocation error of the reactive scheme is not affected by the prediction window length and remains consistent. For the predictive scheme, the resource allocation error is minimized when the prediction window length is 3 cycles. This suggests that in the experimental scenario, there is the highest correlation between the UE's motion and its motion state within the previous 1.2 seconds. Using more historical velocity data, on the other hand, will decrease the accuracy of the prediction.

Velocity of UEs. We select the optimal parameter values for the following experiments. The length of the scheduling cycle is set to 0.4 seconds, and the prediction window length is three cycles. The UE's velocity is set to 1 m/s, 2 m/s, 3 m/s, 5 m/s, and 8 m/s to comprehensively analyze low-speed scenarios and potential high-speed cases. We analyze the trajectory tracking error of the UE during uniform motion, and the results are shown in Figure 14(c). Since the analysis focuses on uniform motion, the velocity prediction algorithm could not exert its effectiveness, resulting in similar performance between the predictive scheduling scheme and the reactive scheduling

scheme. As the velocity increases, the tracking error remains low and stable in the first four experiments. However, at 8 m/s, a noticeable increase in error occurs. Despite this increase, the error remains within an acceptable range.

Under speeds below 2 m/s, the tracking error remains at a low level of 65 cm, indicating the effectiveness of the resource scheduling. For the last case with extremely high speed, we note that such a scenario is rare. We then analyze the resource scheduling process in this case. Due to the 5G frame structure of "DDDDDDDSUU," the last three consecutive slots are allocated for scheduling SRS signals. In high-sampling-rate scenarios, overly dense SRS scheduling may result in neighboring SRS signals being repeatedly allocated, limiting the improvement in sensing performance. Additionally, due to the dynamic changes in signal arrival path during high-speed movement, the sensing performance drops. In conclusion, the results show that the system adapts well to different velocities.

Acceleration of UEs. We set the UE to speed up from 0 m/s to 2 m/s with different accelerations and analyze the trajectory tracking error during the acceleration process. The results are shown in Figure 14(d). As the UE's acceleration increases, both reactive and predictive schemes show an increase in tracking error, with the increase being more gradual in the predictive scheme. The reactive scheme has a scheduling delay of one cycle. Therefore, as the UE's acceleration increases, the delay leads to an increase in tracking error. The predictive scheme can better adapt to motion with a constant acceleration.

4.6 ElaSe Overhead

ElaSe introduces additional overhead to 5G vRAN, which can be divided into the following parts:

(1) **Algorithmic Overhead:** This overhead primarily comes from the following modules: the user state recognition algorithm; the resource allocation algorithm in the resource scheduling module; and the state prediction algorithm used to eliminate scheduling delays.
(2) **RAN Intelligent Controller Scheduling Overhead:** We use the open-source RIC of FlexRAN to transmit SRS scheduling information to the protocol stack. During scheduling, this process incurs a certain degree of scheduling overhead. However, since we employ a near-real-time RIC, the scheduling delay remains below 10 ms.

4.6.1 *Analysis of Algorithmic Computational Complexity.*

(1) **User State Recognition Algorithm:** The primary computational cost stems from the Short-Time Fourier Transform. When the input signal length is N and the window length is M, the computational complexity of the STFT can be approximated as $O(N \cdot logM)$. In our implementation, the sampling frequency of CSI is set to 100 Hz, hence $N \leq 100$.
(2) **Resource Allocation Algorithm:** This primarily involves calculating the weight of each target and aggregating the allocated resources. The computational complexity of calculating each individual weight is a constant C. Therefore, the overall computational complexity of the resource allocation algorithm is $O(C)$.
(3) **State Prediction Algorithm:** The Kalman filtering algorithm is employed. As shown in Table 1, its computational complexity is $O(d^3)$, where the dimension d = 2.
(4) **Summary:** In a scenario with n targets, the total computational complexity of the algorithm is the sum of the above three components. The overall complexity grows linearly with the number of targets n, and can be expressed as:$O(n \cdot (O(N \cdot logM) + O(C) + O(d^3))$.

4.6.2 Time Latency. Our algorithm is executed on a computer equipped with an i9-13900K CPU and 16GB of memory. In our experiments, when the number of targets is 100, the total runtime of the aforementioned algorithm is less than 10 ms. Combining the algorithmic overhead with

the scheduling overhead of the RIC, the overall delay introduced by ElaSe does not exceed 20 ms, which is approximately equivalent to the time span of two 5G frames. Compared with scheduling cycles at the 400 ms level, this latency is negligible.

4.7 Summary of Evaluation

Based on the above evaluations on ElaSe, the following summary can be drawn:

(1) The median trajectory tracking error of ElaSe is 68.69 cm, which is 34% lower than that of the baseline in realistic indoor settings.
(2) Compared with the scheme without any schedule, the resource allocation error of ElaSe is 92% lower, achieving more effective use of sensing resources.
(3) Compared with the proportional fair algorithm, the weight-based allocation algorithm performs better in handling resource oversubscription, achieving a 10.6% reduction in trajectory tracking error.
(4) Length of the scheduling cycle and length of the prediction window influence ElaSe's performance of the velocity prediction differently, but they don't influence the efficacy of the prediction. In the worst case, the resource allocation error of the predictive scheme amounts to 63.68% of that of the reactive scheme.
(5) When the UE moves at different velocities, the system maintains a consistent level of trajectory tracking error. The trajectory tracking error increases as the acceleration of UEs increases, but the prediction scheme can mitigate this upward trend.

5 Discussion and Future Works

In this section, we discuss practical issues concerning the applicability and efficacy of ElaSe and propose corresponding future research directions.

5.1 Applying ElaSe to Other Sensing Cases

In the cases of localization and tracking, velocity is considered as an effective metric for describing the dynamic demands of UEs. However, not all cases can apply velocity as a general metric. For example, in gesture recognition or fall detection, the spectral characteristics of the target have been proven to be an effective indicator. In system design, we particularly pay attention to this point and decouple the user state recognition and SRS resource scheduling modules by using RS in different time slots. This loose coupling ensures the effectiveness of resource scheduling and also provides flexibility for choosing new metrics when applying ElaSe to other sensing cases. When dealing with new sensing cases, it is only necessary to adjust the metrics based on state recognition module. This approach even supports scenarios where UEs perform different sensing tasks concurrently.

5.2 Combining Multi-Stream 5G Reference Signals

First, ElaSe uses the uplink SRS signals only mainly because of the inaccessibility of the CSI processing module of the commercial 5G UEs. The first category of extension is to utilize the uplink DMRS. Since that the DMRS symbols only experience one more layer of precoding than the SRS symbols, successive works can convert the DMRS-based CSI into the SRS-based CSI and extend the total amount of sensing resources. Besides, successive works can utilize the link symmetricity and use the downlink CSI-RS and DMRS signals provided by white-box UE devices for the further extension. Note that such multi-stream extensions are quite straightforward to be implemented in ElaSe with our defined scheduling interface and logic. Challenges like handling the dynamics

caused by the binding between DMRS to data streams and quantifying the quality of multi-channel sensing signals before the integration need to be further addressed.

6 Related Works

6.1 Wireless Sensing with LTE/5G

Nowadays, ISAC with LTE and 5G has attracted growing research attention. Many works model the localization errors, discuss the impacts of modulation methods and sensing algorithms on the errors, and conduct extensive simulation-based evaluations [19, 39, 40]. Besides, Feng et al. propose the interference cancellation [47] and the noise reduction [46] algorithms to address the practical issues when applying traditional signal processing based solutions like AoA/DoA in real-world applications like the pedestrian tracking. Except for improving traditional algorithms, several works utilize classic machine learning algorithms and emerging deep learning algorithms to further reduce the sensing errors [23] and defend the replay attacks [10]. Last, recent innovations in the sensing infrastructure like the uniformed linear array antenna [28], the multi-cell RS integration [29] and multi-source RS integration [3] manage to bring the ISAC's capability to a new level.

Although these works have tried their best to reduce the sensing errors and some have claimed the feasible extension in the multi-target sensing scenarios, they only use the vanilla RS signals with default settings. This prior condition limits their ability to fairly allocate RS resources when availability is constrained. Different from them, ElaSe dynamically recognizes the resource demand of different targets and elastically schedules the limited RS resources among them, which is in parallel with the related advances in the algorithm optimization and the infrastructure upgrading.

6.2 Multi-Target Wireless Sensing

Various wireless signals exhibit the ability to sense multiple targets simultaneously, such as WiFi signals [18, 34, 45], acoustic signals [33, 35–37], mmWave signals [49, 50], and so on. WiPolar [34] leverages the different polarization of reflected signals to accurately separate the multipaths from different targets, which, in turn, allows it to track them simultaneously. Symphony [36] exploits the layout of the microphone array to distinguish acoustic signals from different targets along different paths as well as signals from the same target, then calculates their locations concurrently. Hawkeye [5] assigns different mmWave backscatter tags with on-off keying frequencies as their unique signatures, thus the radar can identify and localize hundreds of tags at on shot by analyzing their modulation patterns.

Despite their inspiring abilities of sensing multiple targets, all usable resources (e.g., frequency) are employed in the existing works without a resource allocation mechanism. In contrast, ElaSe elastically allocates sensing resources among the multiple targets to achieve better scalability.

6.3 Resource Scheduling in vRAN

In terms of resource scheduling in vRAN, existing works focus on the following aspects:

— **Resource slicing and allocation.** Many studies focus on how to improve the performance of vRAN through intelligent resource slicing and allocation strategies. RadioSaber [8] proposes a RAN slicing strategy that enables channel-aware resource allocation at both the inter-slice scheduler and the enterprise scheduler. It can achieve 17%–72% better throughput than other RAN slicing works which perform channel-agnostic inter-slice scheduling. Bonati et al. [7] use deep reinforcement learning agents to achieve closed-loop integration of data analysis and network control, optimizing the scheduling strategies of network slices. These works usually meet the service quality requirements of different users and

applications, but mainly focus on the needs of communication applications, without involving the scheduling of sensing resources.

— **Real-time RAN control.** Some works are dedicated to developing a near real-time RIC framework that dynamically manages network resources through various xApps. This type of research typically focuses on real-time decision-making in wireless environments, such as beamforming, interference management, and channel estimation. EdgeRIC [20] presents a real-time RIC co-located with the distributed unit. It can access RAN and application-level information to execute AI-optimized and other policies in less than 1ms. Janus [9] also aims at real-time measurement and control of the RAN. Janus extends eBPF to allow third-parties to load arbitrary codelets inline in the RAN functions in a provably safe manner. It is integrated directly with each distributed unit, making it vendor-specific. These research results have achieved real-time control of RAN resources and can be easily adjusted for managing sensing resources (e.g., RS) in RAN.

Although existing works have made some progress in resource slicing and real-time control, they typically don't address the scheduling of sensing resources. In contrast, ElaSe proposes the first framework for scheduling SRS in 5G vRAN. Experimental results demonstrate that ElaSe enhances the effectiveness and quality of multi-target sensing tasks.

7 Conclusion

In this article, we present the first sensing resource scheduler in 5G networks, ElaSe, to schedule SRS resources for UEs elastically and in time. ElaSe benefits from the programmability of 5G vRAN and utilizes the interface of its protocol stack. It uses the velocity of UEs to represent the time-variant sensing demands. ElaSe saves sensing resources by minimizing the number of used sensing slots to preserve communication capabilities. It adopts a predictive scheduling scheme to eliminate scheduling delays and uses the weight-based allocation algorithm to handle cases of resource oversubscription better. Extensive experiments under real-world scenarios show that ElaSe can schedule sensing resources appropriately in different situations, achieving small trajectory tracking errors and resource allocation errors.

References

[1] 2023/10/20. 3GPP TS 38.211 Release 16. Retrieved May 01, 2025 from https://www.etsi.org/deliver/etsi_ts/138200_138299/138211/16.02.00_60/ts_138211v160200p.pdf

[2] 2023/10/20. RIC Specifications from O-RAN Alliance. Retrieved May 01, 2025 from https://orandownloadsweb.azurewebsites.net/specifications

[3] Ali A. Abdallah, Joe Khalife, and Zaher M. Kassas. 2023. Exploiting on-demand 5G downlink signals for opportunistic navigation. *IEEE Signal Processing Letters* 30 (2023), 389–393.

[4] Bharat Agarwal, Mohammed Amine Togou, Marco Ruffini, and Gabriel-Miro Muntean. 2022. QoE-driven optimization in 5G O-RAN-Enabled HetNets for enhanced video service quality. *IEEE Communications Magazine* 61, 1 (2022), 56–62.

[5] Kang Min Bae, Hankyeol Moon, Sung-Min Sohn, and Song Min Kim. 2023. Hawkeye: Hectometer-range subcentimeter localization for large-scale mmWave backscatter. In *Proceedings of the ACM MobiSys*.

[6] Carlos Baena, María Hervás-Gutiérrez, Eduardo Baena, Javier Villegas, Raquel Barco, and Sergio Fortes. 2023. Assessing the impact of computational resources to the quality of experience provided by vRANs. *IEEE Access* 11 (2023), 102944–102948.

[7] Leonardo Bonati, Salvatore D'Oro, Michele Polese, Stefano Basagni, and Tommaso Melodia. 2021. Intelligence and learning in O-RAN for data-driven NextG cellular networks. *IEEE Communications Magazine* 59, 10 (2021), 21–27.

[8] Yongzhou Chen, Ruihao Yao, Haitham Hassanieh, and Radhika Mittal. 2023. Channel-Aware 5G RAN slicing with customizable schedulers. In *Proceedings of the USENIX NSDI*.

[9] Xenofon Foukas, Bozidar Radunovic, Matthew Balkwill, and Zhihua Lai. 2023. Taking 5G RAN analytics and control to a new level. In *Proceedings of the ACM Mobicom*.

[10] Kaixuan Gao, Huiqiang Wang, Hongwu Lv, and Pengfei Gao. 2023. Your locations may be lies: Selective-PRS-Spoofing attacks and defence on 5G NR positioning systems. In *Proceedings of the IEEE INFOCOM*.

[11] Gines Garcia-Aviles, Andres Garcia-Saavedra, Marco Gramaglia, Xavier Costa-Perez, Pablo Serrano, and Albert Banchs. 2021. Nuberu: Reliable RAN virtualization in shared platforms. In *Proceedings of the ACM Mobicom*.
[12] Andres Garcia-Saavedra and Xavier Costa-Perez. 2021. O-RAN: Disrupting the virtualized RAN ecosystem. *IEEE Communications Standards Magazine* 5, 4 (2021), 96–103.
[13] Xiuzhen Guo, Yuan He, Xiaolong Zheng, Liangcheng Yu, and Omprakash Gnawali. 2018. ZigFi: Harnessing channel state information for cross-technology communication. In *Proceedings of the IEEE INFOCOM*.
[14] Tianmeng Hang, Yue Zheng, Kun Qian, Chenshu Wu, Zheng Yang, Xiancun Zhou, Yunhao Liu, and Guilin Chen. 2019. WiSH: WiFi-based real-time human detection. *Tsinghua Science and Technology* 24, 5 (2019), 615–629.
[15] Chengkun Jiang, Junchen Guo, Yuan He, Meng Jin, Shuai Li, and Yunhao Liu. 2020. mmVib: Micrometer-level vibration measurement with mmWave radar. In *Proceedings of the ACM MobiCom*.
[16] Haotian Jiang, Jiacheng Zhang, Xiuzhen Guo, and Yuan He. 2021. Sense me on the ride: Accurate mobile sensing over a LoRa backscatter channel. In *Proceedings of the ACM Sensys*.
[17] Zhiping Jiang, Wei Xi, Xiangyang Li, Shaojie Tang, Jizhong Zhao, Jinsong Han, Kun Zhao, Zhi Wang, and Bo Xiao. 2014. Communicating is crowdsourcing: Wi-Fi indoor localization with CSI-based speed estimation. *Journal of Computer Science and Technology* 29, 4 (2014), 589–604.
[18] Chitra R. Karanam, Belal Korany, and Yasamin Mostofi. 2019. Tracking from one side: Multi-person passive tracking with WiFi magnitude measurements. In *Proceedings of the ACM/IEEE IPSN*.
[19] Ryan Keating, Mikko Säily, Jari Hulkkonen, and Juha Karjalainen. 2019. Overview of positioning in 5G new radio. In *Proceedings of the IEEE ISWCS*.
[20] Woo-Hyun Ko, Ushasi Ghosh, Ujwal Dinesha, Raini Wu, Srinivas Shakkottai, and Dinesh Bharadia. 2024. EdgeRIC: Empowering real-time intelligent optimization and control in NextG cellular networks. In *Proceedings of the USENIX NSDI*.
[21] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. 2015. SpotFi: Decimeter level localization using WiFi. In *Proceedings of the ACM SIGCOMM*.
[22] Bo Liang, Purui Wang, Renjie Zhao, Heyu Guo, Pengyu Zhang, Junchen Guo, Shunmin Zhu, Hongqiang Harry Liu, Xinyu Zhang, and Chenren Xu. 2023. RF-Chord: Towards deployable RFID localization system for logistic networks. In *Proceedings of the USENIX NSDI*.
[23] Zhaoliang Liu, Liang Chen, Xin Zhou, Zhenhang Jiao, Guangyi Guo, and Ruizhi Chen. 2023. Machine learning for time-of-arrival estimation with 5G signals in indoor positioning. *IEEE Internet of Things Journal* 10, 11 (2023), 9782–9795.
[24] Zhihong Luo, Qiping Zhang, Yunfei Ma, Manish Singh, and Fadel Adib. 2019. 3D backscatter localization for fine-grained robotics. In *Proceedings of the USENIX NSDI*.
[25] Xin Na, Xiuzhen Guo, Yuan He, and Rui Xi. 2021. Wi-attack: Cross-technology impersonation attack against iBeacon services. In *Proceedings of the IEEE SECON*.
[26] Xin Na, Xiuzhen Guo, Zihao Yu, Jia Zhang, Yuan He, and Yunhao Liu. 2023. Leggiero: Analog WiFi backscatter with payload transparency. In *Proceedings of the ACM MobiSys*.
[27] Kai Niu, Xuanzhi Wang, Fusang Zhang, Rong Zheng, Zhiyun Yao, and Daqing Zhang. 2022. Rethinking doppler effect for accurate velocity estimation with commodity WiFi devices. *IEEE Journal on Selected Areas in Communications* 40, 7 (2022), 2164–2178.
[28] Mengguan Pan, Peng Liu, Shengheng Liu, Wangdong Qi, Yongming Huang, Xiaohu You, Xinghua Jia, and Xiaodong Li. 2022. Efficient joint DOA and TOA estimation for indoor positioning with 5G picocell base stations. *IEEE Transactions on Instrumentation and Measurement* 71 (2022), 1–19.
[29] Rui Peng, Yafei Tian, and Shengqian Han. 2023. ICI-free channel estimation and wireless gesture recognition based on cellular signals. *IEEE Wireless Communications Letters* 12, 12 (2023), 2088–2092.
[30] Robert Schmidt, Mikel Irazabal, and Navid Nikaein. 2021. FlexRIC: An SDK for next-generation SD-RANs. In *Proceedings of the ACM CoNEXT*.
[31] Elahe Soltanaghaei, Avinash Kalyanaraman, and Kamin Whitehouse. 2018. Multipath triangulation: Decimeter-level WiFi localization and orientation with a single unaided receiver. In *Proceedings of the ACM MobiSys*.
[32] Yimiao Sun, Yuan He, Jiacheng Zhang, Xin Na, Yande Chen, Weiguo Wang, and Xiuzhen Guo. 2023. BIFROST: Reinventing WiFi signals based on dispersion effect for accurate indoor localization. In *Proceedings of the ACM SenSys*.
[33] Yimiao Sun, Weiguo Wang, Luca Mottola, Ruijin Wang, and Yuan He. 2022. AIM: Acoustic inertial measurement for indoor drone localization and tracking. In *Proceedings of the ACM SenSys*.
[34] Raghav H. Venkatnarayan, Muhammad Shahzad, Sangki Yun, Christina Vlachou, and Kyu-Han Kim. 2020. Leveraging polarization of WiFi signals to simultaneously track multiple people. *Proceedings of ACM IMWUT* 4, 2 (2020), 45:1–45:24.
[35] Weiguo Wang, Yuan He, Meng Jin, Yimiao Sun, and Xiuzhen Guo. 2023. Meta-Speaker: Acoustic source projection by exploiting air nonlinearity. In *Proceedings of the ACM MobiCom*.

[36] Weiguo Wang, Jinming Li, Yuan He, and Yunhao Liu. 2020. Symphony: Localizing multiple acoustic sources with a single microphone array. In *Proceedings of the ACM SenSys*.
[37] Weiguo Wang, Luca Mottola, Yuan He, Jinming Li, Yimiao Sun, Shuai Li, Hua Jing, and Yulei Wang. 2022. MicNest: Long-range instant acoustic localization of drones in precise landing. In *Proceedings of the ACM SenSys*.
[38] Weiguo Wang, Xiaolong Zheng, Yuan He, and Xiuzhen Guo. 2019. AdaComm: Tracing channel dynamics for reliable cross-technology communication. In *Proceedings of the IEEE SECON*.
[39] Zhiqing Wei, Hanyang Qu, Yuan Wang, Xin Yuan, Huici Wu, Ying Du, Kaifeng Han, Ning Zhang, and Zhiyong Feng. 2023. Integrated sensing and communication signals towards 5G-A and 6G: A survey. *IEEE Internet of Things Journal* 10, 13 (2023), 11068–11092.
[40] Zhiqing Wei, Yuan Wang, Liang Ma, Shaoshi Yang, Zhiyong Feng, Chengkang Pan, Qixun Zhang, Yajuan Wang, Huici Wu, and Ping Zhang. 2022. 5G PRS-based sensing: A sensing reference signal approach for joint sensing and communication system. *IEEE Transactions on Vehicular Technology* 72, 3 (2022), 3250–3263.
[41] Chenshu Wu, Feng Zhang, Yuqian Hu, and K. J. Ray Liu. 2021. GaitWay: Monitoring and recognizing gait speed through the walls. *IEEE Transactions on Mobile Computing* 20, 6 (2021), 2186–2199.
[42] Zhiqiang Xiao and Yong Zeng. 2022. An overview on integrated localization and communication towards 6G. *SCIENCE CHINA Information Sciences* 65, 3 (2022), 131301.
[43] Binbin Xie, Yuqing Yin, and Jie Xiong. 2021. Pushing the limits of long range wireless sensing with LoRa. *Proceedings of ACM IMWUT* 5, 3 (2021), 134:1–134:21.
[44] Dongzhu Xu, Anfu Zhou, Guixian Wang, Huanhuan Zhang, Xiangyu Li, Jialiang Pei, and Huadong Ma. 2022. Tutti: Coupling 5G RAN and mobile edge computing for latency-critical video analytics. In *Proceedings of the ACM MobiCom*.
[45] Leiyang Xu, Xiaolong Zheng, Xiangyuan Li, Yucheng Zhang, Liang Liu, and Huadong Ma. 2022. WiCAM: Imperceptible adversarial attack on deep learning based WiFi sensing. In *Proceedings of the IEEE SECON*.
[46] Y. Feng, Y. Xie, D. Ganesan, and J. Xiong. 2022. LTE-based low-cost and low-power soil moisture sensing. In *Proceedings of the ACM SenSys*.
[47] Y. Feng, Y. Xie, D. Ganesan, and J. Xiong. 2021. LTE-based pervasive sensing across indoor and outdoor. In *Proceedings of the ACM SenSys*.
[48] Gu Yu, Meng Wang, Peng Zhao, Yantong Wang, Hao Zhou, Yusheng Ji, and Celimuge Wu. 2022. SpiroFi: Contactless pulmonary function monitoring using WiFi signal. In *Proceedings of the IEEE/ACM IWQoS*.
[49] Jia Zhang, Xin Na, Rui Xi, Yimiao Sun, and Yuan He. 2023. mmHawkeye: Passive UAV detection with a COTS mmWave radar. In *Proceedings of the IEEE SECON*.
[50] Jia Zhang, Rui Xi, Yuan He, Yimiao Sun, Xiuzhen Guo, Weiguo Wang, Xin Na, Yunhao Liu, Zhenguo Shi, and Tao Gu. 2023. A survey of mmWave-Based human sensing: Technology, platforms and applications. *IEEE Communications Surveys & Tutorials* 25, 4 (2023), 2052–2087.
[51] Xiaolong Zheng, Jiliang Wang, Longfei Shangguan, Zimu Zhou, and Yunhao Liu. 2017. Design and implementation of a CSI-based ubiquitous smoking detection system. *IEEE/ACM Transactions on Networking* 25, 6 (2017), 3781–3793.
[52] Xiaolong Zheng, Kun Yang, Jie Xiong, Liang Liu, and Huadong Ma. 2024. Pushing the limits of WiFi sensing with low transmission rates. *IEEE Transactions on Mobile Computing* 23, 11 (2024), 10265–10279.
[53] Xiang Zou, Jianwei Liu, Jinsong Han, Wei Xi, and Zhi Wang. 2023. WiHunter: Enabling real-time small object detection via wireless sensing. In *Proceedings of the IEEE/ACM IWQoS*.

Received 31 December 2024; revised 3 April 2025; accepted 5 May 2025