Handling Failures in Secondary Radio Access Failure Handling in Operational 5G Networks

Yanbing Liu, *Graduate Student Member, IEEE*, and Chunyi Peng, *Senior Member, IEEE*

***Abstract*—In this work, we conduct a measurement study with three US operators to reveal three types of problematic failure handling on secondary radio access which have not been reported before. Compared to primary radio access failures, secondary radio access failures do not hurt radio access availability but significantly impact data performance, particularly when 5G is used as secondary radio access to boost throughput. Improper failure handling results in significant throughput loss, which is unnecessary in most instances. We then pinpoint the root causes behind these three types of problematic failure handling. When 5G provides higher throughput, failures are more likely to be falsely triggered by a specific event, causing the User Equipment (UE) to unnecessarily lose well-performing 5G connections. Moreover, after failures, the recovery of secondary radio access may fail due to inconsistent parameter settings or be delayed due to missing specific signaling fields. To address these issues, we propose SCGFailure Manager (SFM), a solution to optimize the detection and recovery of secondary radio access failures. Our evaluation results demonstrate that SFM can effectively avoid 60%-80% of problematic failure handling and double throughput in more than half of failure instances.**

***Index Terms*—5G, cellular network, SCGFailure, SCGFailure manager (SFM), secondary cell group (SCG).**

I. Introduction

HANDLING radio access failures is essential to cellular network reliability, availability and performance. When the radio link (RL)[1] between a mobile device and its serving cell (also known as a base station) fails to transmit packets in the air, the ongoing data/voice sessions are interrupted until this radio link failure (RLF) is recovered (e.g., by another RL that works).

Handling radio access failures is more complex and harder, as cellular networks advance from 3G/4G to 4.5G/5G and beyond. In a 3G/4G network, a radio access failure is a *YES-or-NO* problem; Radio access is available (or unavailable) when the used RL does not fail (or fails). This is because 3G/4G uses a single RL to serve a mobile device. The problem turns more complicated as 4.5G/5G increases the number of active RLs from 1 to N ($N \geq 1$, mostly $N \gg 1$), through two advanced radio access technologies: carrier aggregation [1], [2] and dual connectivity [3], [4].[2] The former uses a group of serving cells, which was first adopted by 4.5G LTE-advanced [1]; The latter uses two cell groups, which was launched by 5G [3]. Specifically, each serving cell uses one RL over one frequency channel as a basic unit to offer radio access. All the serving cells are grouped into *Master Cell Group* (MCG) and *Secondary Cell Group* (SCG), based on their radio access technologies (RATs, here, 4G[3] and 5G) [3]. Each group uses carrier aggregation to combine one primary cell (PCell) and several secondary cells of the same RAT [2]. As a result, 5G aggregates radio frequency channels used by all active RLs over 5G and 4G/4.5G, thereby utilizing much wider radio frequency spectrum to boost network performance. Unsurprisingly, 5G is often much faster than 4G/4.5G, up to several hundreds of Mbps [5].

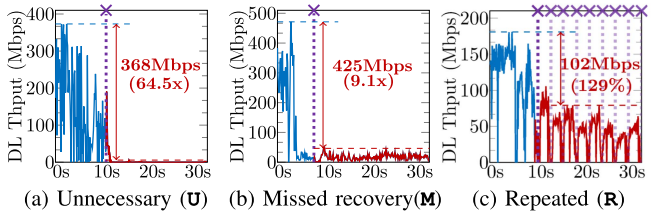


Fig. 1. Real-world instances of three types of "problematic" SCGFailure handling (**U**, **M**, **R**).

Radio access failures are handled at two levels: *logic* and *physical*. The above logic level is managed by radio resource control (RRC), which is responsible for establishing and maintaining a logic channel (namely, an active RRC connection) to transfer user traffic. Its connection state is still *YES-or-NO*, say, active/connected or idle. This logic channel is provisioned through *physical* RLs. 5G uses only MCG[4] to manage the logic RRC connection, and both MCG and SCG for physical radio access to mobile devices: MCG for primary radio access and SCG for secondary and opportunistic radio access.

In this work, we examine how 5G handles secondary radio access failures. Such failure is officially termed as *SCGFailure*, which was introduced in Release-15 of the 3rd Generation

Received 19 June 2024; revised 3 September 2024; accepted 25 September 2024. Date of publication 10 October 2024; date of current version 9 January 2025. This work was supported in part by NSF under Grant CNS-1750953 and Grant CNS-2112471. Recommended for acceptance by D. Niyato. *(Corresponding author: Chunyi Peng.)*

The authors are with the Department of Computer Science, Purdue University, West Lafayette, IN 47907 USA (e-mail: liu3098@purdue.edu; chunyi@purdue.edu).

Digital Object Identifier 10.1109/TMC.2024.3477462

[1]The abbreviations and their full names are summarized in Table II.

[2]Dual-connectivity has been extended to multi-connectivity in the recent 3GPP standard specification [4]. In this work, we focus on dual-connectivity because multi-connectivity has not been observed in operational cellular networks and all the findings are conceptually applicable to multi-connectivity.

[3]In the rest of the paper, 4G is used to represent all 4G variants, including LTE (4G), LTE-Advanced (4.5G) and LTE-Advanced Pro (4.75G). All US operators support 4.5G/4.75G.

[4]Actually, the PCell of MCG manages the RRC connection.

TABLE I
SUMMARY OF OUR MAIN FINDINGS ON THREE TYPES OF "PROBLEMATIC" SCGFAILURE HANDLING IN OUR REALITY CHECK

	Description	Performance Impact	Root Cause	OPs	Main Results
U	Unnecessary handling	Performance unnecessarily drops	Retransmission triggering event improperly configured	**A**, **T**, **V**	Figure 1a, 3, 8, 12, 16a, 17a, 18a, 18b
M	Missed recovery	Long-time poor performance	No piggybacked measurement report to recover the failure	**A**, **T**, **V**	Figure 1b, 11, 14, 16b, 17b, 18c, 18d
R	Repeated failures	Long-time performance fluctuation	Random access failures due to too low SCG Addition threshold	**A**, **T**, **V**	Figure 1c, 9, 10, 13, 16c, 17c, 18e, 18f

TABLE II
SUMMARY OF ABBREVIATIONS IN THIS PAPER

Abbr.	Full name	Abbr.	Full name
RL	Radio Link	RLF	Radio Link Failure
PCell	Primary Cell	SCell	Secondary Cell
MCG	Master Cell Group	SCG	Secondary Cell Group
RRC	Radio Resource Control		
U	Unnecessary Failure Handling		
M	Missed Failure Recovery		
R	Repeated Failures		
SCGFailure	Secondary Radio Access Failure		
SFM	SCGFailure Manager		

Partnership Project (3GPP), the first set of 5G standards [6]. An SCGFailure occurs when one or more RLs used by SCG fail but the RLs by MCG not. Therefore, the device, upon detecting an SCGFailure, is still able to report the detected failure to MCG and invoke a RRC procedure to recover the SCGFailure. In principle, SCGFailures do not harm access availability but impact data performance.

In this work, we are particularly interested in characterizing and understanding SCGFailure handling in operational cellular networks and demystifying *"problematic"* failure handling. We indeed observe *"problematic"* failure handling with three major US operators (AT&T, Verizon and T-Mobile, short as **A**, **V** and **T** afterwards). Fig. 1 gives three real-world instances per type observed in our study (each SCGFailure marked as $\times$), which all result in substantial performance loss. In this work, we have unveiled three types of *"problematic"* failure handling:

- *Unnecessary failure handling:* A SCGFailure is falsely detected and reported, resulting in the unnecessary removal of SCG RLs which can offer good performance (Fig. 1(a)). Here, the peak downlink throughput shrinks from 373.2 Mbps to 5.7 Mbps, losing 368 Mbps ($368/5.7 = 64.5\times$).
- *Missed failure recovery:* A true SCGFailure is detected but not recovered in presence of suitable RLs, which results in significant performance degradation (Fig. 1(b)). Here, the peak rate decreases by one order of magnitude (471.3 Mbps $\rightarrow$ 46.5 Mbps).
- *Repeated failures:* A true SCGFailure is recovered but the recovery does not last long (Fig. 1(c)); SCGFailures are frequently repeated every a few seconds because the failed RL is used again for the recovery. Data throughput not only oscillates but also greatly declines by 129% (about 102 Mbps).

We elaborate these three problematic cases of SCGFailure handling in Section III. Most importantly, we notice that such failure handling is "*problematic*", not because current practice does not follow the standard procedures. Instead, current practice conforms to 3GPP standards but still suffers *significant-but-unnecessary* performance degradation. We then pinpoint the root causes behind these three types of problematic failure handling. First, the SCGFailure detection mechanism is not aware of the impact of user traffic load and parameter settings. As the result, when the UE's throughput is high (e.g., $>$ 100 Mbps) and the parameter settings (retransmission timer) are aggressive, SCGFailure will be unnecessarily triggered by a specific event when SCG is functioning normally. Second, during SCG recovery, even if UE has identified available candidate cells through measurement, if the measurement report of these cells is not piggybacked in the SCGFailure message, UE cannot report the candidate cell information to the network side, which blocks the SCG recovery process. Finally, when the Reference Signal Received Power (RSRP) threshold of SCG addition is set lower than the threshold of random access failure, UE may be trapped in a loop of random access failure and SCG addition. Such loop leads to repeated SCGFailures, resulting in an extremely unstable link environment. We quantitatively characterize the prevalence and performance impacts of these three types of problematic SCGFailure handling. We find that more than half of SCGFailures are handled improperly, with throughput dropping by more than 50% in these instances. Table I summarizes main findings of our measurement study.

To address problematic SCGFailure handling, we propose a device-side solution SCGFailure Manager (SFM) in Section IV. SFM incorporates three key modules to enhance the detection and recovery process of SCGFailures. First, to avoid false detection of SCGFailure, SFM provides traffic-aware SCGFailure detection by utilizing recent traffic information on the UE side. Second, during the recovery stage, when the normal SCG recovery is blocked, SFM offers device-side target cell selection for UE to perform SCG recovery autonomously. Finally, SFM adapts the RSRP threshold of SCG addition to prevent repeated SCGFailures by forcing an early stop. We employ a trace-driven evaluation using what-if study to assess the performance of SFM across all SCGFailure instances in our dataset. With SFM, UE can eliminate more than half of problematic SCGFailure instances. This brings a throughput gain of over 100% in at least half of SCGFailure instances.

Release: Our SCGFailure datasets are available at [7].

II. BACKGROUND AND METHODOLOGY

We first introduce necessary background on SCGFailure and its handling, and then present the methodology and datasets used in this study.

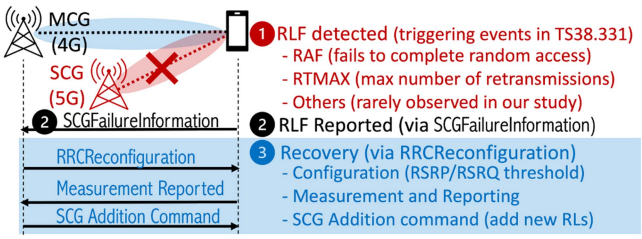


Fig. 2. A typical flow of SCGFailure handling.

TABLE III
TRIGGERING EVENTS FOR SCGFAILURE DETECTION [3]

	Event Description	Parameter(s)
RAF	T304 expires before random access to SCG completes.	T304
RTMAX	Maximum number of retransmissions has been reached.	t-PollRetransmit, rlc-MaxNumRetx
SYNC	Cell synchronization fails during SCG cell addition.	N/A
T310	T310 expires with N310 consecutive out-of-sync indications.	T310, N310
CONF	SCG RRC reconfiguration fails.	N/A
SRB	SRB3 integrity check fails.	N/A

A. SCGFailure Primer

Fig. 2 depicts a typical flow of handling a SCGFailure, which is regulated in 3GPP standards [3], [4]. In this flow, the primary radio access through MCG works at all time. Nowadays, 5G networks use dual-connectivity over two RATs (here, 5G and 4G) to offer radio access to user equipment (UE) [4]. There are three forms of dual-connectivity: (1) 4G MCG + 5G SCG, (2) 5G MCG + 4G SCG, and (3) 5G MCG + 5G SCG. In our measurement study, we see that 4G MCG + 5G SCG is dominant in the US, using 4G for MCG and 5G for SCG, which is the focus of this work.

The handling of SCGFailure starts with detecting a radio link failure (RLF) (❶). 3GPP defines six triggering events (Table III) to detect a RLF [3]. The UE keeps monitoring each active RL and detects a SCGFailure when the RL used by SCG experiences one of the following events: (1) RAF (random access to the SCG cell fails), (2) RTMAX (the maximum number of retransmissions is reached), (3) SYNC (cell synchronization fails), (4) T310 (timer T310 expires with many out-of-sync indications), (5) CONF (RRC reconfiguration fails), and (6) SRB (integrity check for SRB3 fails). Note that the detection criteria are customized by tunable parameters including timers (e.g., T304, T310, t-PollRetransmit) and counts (e.g., N310, rlc-MaxNumRetx) and to name a few. For instance, multiple random access attempts are allowed within a time period (here, T304), and an RAF event occurs when the timer T304 expires and random access still fails. RTMAX uses two key parameters of t-PollRetransmit and rlc-MaxNumRetx. t-PollRetransmit is a timer to initiate data retransmissions and rlc-MaxNumRetx is the maximum number of retransmissions allowed. An RTMAX event occurs when the number of retransmissions reaches its maximum threshold. All the event details are specified in 3GPP standards [3]. In our reality check (Section III), we see that SCGFailure handling is mainly associated with two events (RAF and RTMAX) and other failure types rarely happen, so we focus on these two dominant events in this work.

Upon detecting a triggering event, the device reports the detected RLF (via a signaling message called SCGFailure-Information) to the network with the detected event as its failure type (❷). The network immediately releases the "failed" RL and invokes a standard procedure to recover this RLF (❸). Specifically, recovery is realized by RRC Reconfiguration Procedure [3], which is used to find and add RLs available and suitable (say, meeting the RSRP/RSRQ requirement). It performs four steps: *configuration, measurement, reporting* and *command* (here, SCG Addition). The criteria for measurement and reporting are configured through several tunable parameters, particularly those RSRP/RSRQ thresholds and offsets used to compare radio quality of serving RLs and candidate RLs. The device needs to measure RSRP/RSRQ of available RLs (namely, candidate cells) and report found new cells in order to replace the failed RL. Finally, the network sends a command to add the new SCG cell(s), namely, adding the new RL(s) to replace the "failed" RL(s) which were released before.

B. Methodology and Datasets

We characterize and analyze real-world SCGFailure instances using three datasets **D1**, **D2** and **D3**. **D1** is a public dataset from our recent 5G measurement study [8]. It was collected over 5G experiments with three US operators in two US cities (Chicago and Indianapolis, total area: 19.8 Km^2) from April 2021 to January 2022 (705 hours in total). **D1** was a general dataset and was not specifically collected for SCGFailure study. During collection of **D1**, we conducted passive tests on random routes and locations, rather than performing customized experiments for SCGFailure study. Therefore, even though we observed that many SCGFailure handling instances in **D1** are questionable, we lack necessary information for more in-depth analysis of these instances, such as cell coverage and data performance at those locations. With new findings and insights gained in **D1**, we conducted a 2-month measurement study from September to October 2023 in West Lafayette, and collected **D2** focusing on problematic failure handling. By conducting repeated controlled experiments on locations with frequent SCGFailures, we obtained the ground truth of cell deployment, radio quality, and data performance at these locations, allowing us to analyze whether each instance of SCGFailure handling is problematic, quantify its impact and identify its root causes. In both **D1** and **D2**, we only run one kind of application bulky file downloading to conduct speedtest. As a supplement, we collected a new dataset **D3** on **T**'s 5G network in West Lafayette. This dataset is used to validate the effectiveness of our solution SFM on two applications, file downloading and file uploading, in Section IV-B.

To collect these three datasets, we equip our UE with an open-source tool, MobileInsight [9], to capture cellular signaling messages. Through an analysis of captured signaling messages, we extract RRC procedures before and after each SCGFailure

TABLE IV
THREE DATASETS (**D1**, **D2**, **D3**) USED IN THIS STUDY (FD = FILE DOWNLOADING, FU = FILE UPLOADING)

	D1	D2	D3
Operators	**A**, **V**, **T**	**A**, **V**, **T**	**T**
Traffic type	FD	FD	FD, FU
Period	04/21–01/22	09/23–10/23	12/23–06/24
Region	19.8 Km^2	5.1 Km^2	1.0 Km^2
Duration	42,300 min	2,640 min	1,650 min
♯ **Connections**	129.8K	7.5K	9.5K
♯ **SCGFailures**	4,275	284	464

instance to identify the underlying logic of SCGFailure handling. We also use `tcpdump` to collect throughput traces to analyze the impact of SCGFailure handling on data performance. Table IV summarizes basic statistics of these three datasets. In **D1**, **D2** and **D3**, there are 129.8K, 7.5K and 9.5K RRC connections each with one or more RLs changed; We observe 4,275, 284 and 464 SCGFailure instances, which account for 3.3%, 3.8% and 4.9% of all instances in these three datasets. We notice that the SCGFailure rate is not very high; It matches with our expectation; Operational cellular networks are largely successful and radio access failures should not be common. The astonishing finding is that more than half SCGFailure handling instances are problematic, which will be elaborated next.

III. REALITY CHECK IN THE US

In this section, we present our measurement study of SCGFailure handling in operational 5G networks with three US operators: **A**, **V** and **T**. We uncover three types of problematic failure handling which have been never reported before: (**U**) unnecessary failure handling, (**M**) missed failure recovery, and (**R**) repeated failures. We characterize their prevalence and performance impacts and analyze their underlying causes.

A. Illustrative Examples

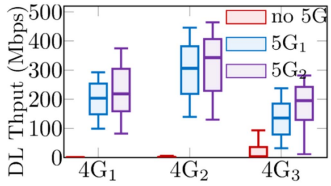
We start with three real-world instances (Fig. 1) to unveil how SCGFailures are exactly (and improperly) handled in reality.

Unnecessary SCGFailure Handling: Fig. 1(a) shows a stationary instance observed in West Lafayette (**D2**) with **T**, which runs 5G over sub-6GHz (< 6GHz). In our study, we see that all three US operators use 4G MCG + 5G SCG, 4G for MCG and 5G for SCG. It is not hard to understand that problematic SCGFailure handling significantly hurts performance as 5G RLs are not properly utilized.

Initially, the device is served by 4G MCG + 5G SCG, achieving high throughput (median: 204 Mbps). The 5G SCG RL is used by a cell $5G_1$ (459@F520110). Here, 459 is its cell ID and F520110 is its channel number, as specified in [2]; F520110 is a 5G channel centered on 2600 MHz with its channel width of 100 MHz. At 10 s, a SCGFailure is detected with an RTMAX event, where the number of continuous retransmissions reaches its maximum (`rlc-MaxNumRetx`: 32) within a short interval (`t-PollRetransmit`: 45 ms). The device reports this detected RLF and the 5G RL is released immediately (though it is still able to offer high data speed). In order to expedite failure recovery, 3GPP recommends piggybacking the RSRP/RSRQ measurements of neighboring SCG cells while reporting the detected RLF [3]. In this instance, no measurement results of available 5G cells are piggybacked. Later at 17s, the device receives a new message `RRCReconfiguration` which configures the device to measure nearby cells over other 5G frequencies (here, F520110 and F125290). Surprisingly, no measurement results of 5G cells are reported despite the presence of four good 5G cells (Fig. 3(a)). As a result, the device loses 5G as its secondary radio access and uses 4G only; Throughput shrinks below 10 Mbps.

	5G Cell	Width (MHz)	RSRP (dBm)
$5G_1$	**459@F520110**	100	-94.6
$5G_2$	**694@F520110**	100	-99.3
$5G_3$	711@F125290	20	-86.5
$5G_4$	55@F125290	20	-90.4

(a) 5G cells with good RSRPs



(b) Throughput of main cellsets

Fig. 3. 5G SCG cells and main cellsets observed at the same location in the example instance of unnecessary failure handling (Fig. 1(a)).

This instance is *"problematic"* because 5G cells with good radio quality and high data throughput (hundreds of Mbps) are present but not used. We run extensive experiments at the same location and observe such 5G cells. Fig. 3(a) lists four 5G cells with good radio quality (medium RSRP > -100 dBm). As long as $5G_1$ or $5G_2$ is used, data throughput is much higher (than no 5G), as shown in Fig. 3(b). $4G_1$, $4G_2$ and $4G_3$ are three 4G cells used by the MCG.

We further examine why problematic failure handling occurs in this instance. It is attributed to two issues: *(1) improper* RLF *detection*, and *(2) no failure recovery*.

First, RLF is falsely detected by an RTMAX event which uses a threshold (`rlc-MaxNumRetx`: 32) and cannot effectively distinguish a SCGFailure under light/normal traffic and a normal use under heavy traffic. Ironically, when 5G SCG provides very high throughput (hundreds of Mbps), it highly likely experiences more than 32 continuous retransmissions even though many more packets are successfully transmitted over the used RL. Evidently, the higher throughput provided by 5G SCG, the higher the likelihood of a false trigger (an RTMAX event indicating the failure of the used RL [3])(or the higher the likelihood of losing high throughput provided by 5G SCG). Later, we will show that it is the dominant source to false RLF detection, which is commonly observed with all instances with unnecessary failure handling with all three US operators in Section III-B.

Second, it is indeed hard to use this single instance to figure out why the failure is not recovered. We thus examine many instances with and without failure recovery to learn what makes a difference. We find that no recovery in presence of suitable 5G RLs is highly correlated with another operation: *no piggybacked measurement reports while reporting the detected* RLF *(via* `SCGFailureInformation`*)*. Interestingly, we find that the subsequent `RRC Reconfiguration Procedure` becomes ineffective if no measurement report is piggybacked. As long as at least one measurement report of any neighboring

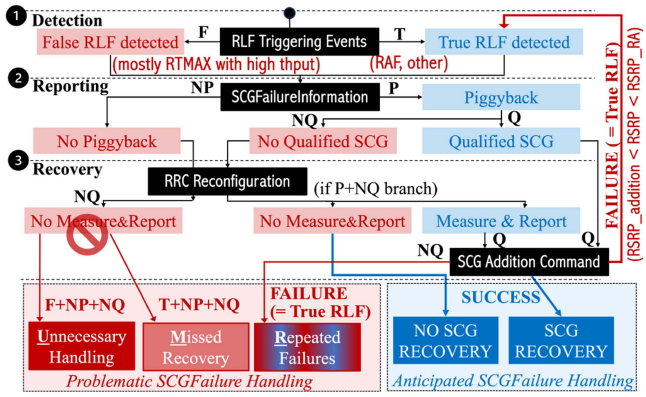


Fig. 4. FSM for problematic SCGFailure handling.

SCG cells is piggybacked, the recovery procedure can proceed: immediately add the qualified SCG cells in the piggybacked report or use the subsequent RRC Reconfiguration Procedure to later add qualified SCG cells which are not included in the piggybacked report (more details in Fig. 4).

We also notice that unnecessary failure handling is not rare at the same location. It is true that SCGFailures are not common; The failure rate is 3.3% (4275 out of 129.8K) observed in **D1**, which ran a field test in two US cities [8]. Otherwise, cellular networks would not be largely successful. However, unnecessary failure handling is not rare where it occurs. At this location, more than 85% of SCGFailures are unnecessary, resulting in huge throughput loss ($>$ 280 Mbps), in spite of various SCG and MCG cells involved.

Missed SCGFailure Recovery: Fig. 1(b) is observed also with **T** in West Lafayette (**D2**), but on a walk route. The main difference from the above instance is that the detected SCGFailure is true. At 3.8s, data throughput drops below 30 Mbps from $\sim$ 300 Mbps. The involved 5G SCG cell is 523@F520110 (a different cell but over the same channel). The RL indeed fails to complete random access to the 5G SCG cell, indicating that the uplink to the network does not work. As a result, the SCGFailure is detected with an RAF event, not with an RTMAX event. The problem lies in no recovery in presence of good RLs. We see two 5G cells (700@F125290, 700@F126270) with good radio quality (RSRP $>$ $-$93 dBm), each of which can yield 100–150Mbps once used. The plot is skipped due to space limit. However, recovery is missed for the same reason: *no measurement reports are piggybacked*, which blocks the reporting of qualified cells. It holds true for most instances with missed recovery.

Repeated SCGFailures: Fig. 1(c) shows a stationary instance with **A**, another US operator observed in **D1**. At the start, the device does get high data throughput of 100–180 Mbps. At 9.2s, the device attempts to add a new SCG cell (634@F174270). F174270 is a 5G channel centered on 871 MHz (sub-6GHz). This 5G cell is measured with RSRP $=$ $-$105 dBm, which is higher than the threshold ($-$110 dBm) needed for SCG addition. Then the device adds this cell but at 9.6s (400 ms later), it detects the RLF with an RAF event (similar to the second instance in Section III-A). As a result, this SCG cell is released. It should not be a problem when the RL with a high RSRP value might fail (here, random access failure). The real problem is that the above process is frequently repeated. At 11.8s, the device attempts to reconnect to the same SCG cell but at 12.2s, the same SCGFailure happens again due to another RAF event. It is repeated for nine times within 20 seconds (9.2s, 30s). It keeps oscillating with two operations: SCG Addition and SCG Removal (due to SCGFailures). As a consequence, the overall throughput drops from 100 –180 Mbps to 0–80 Mbps. Clearly, repeated failures can be avoided if the network avoids the same mistake again and again.

B. Breakdown and Cause Analysis

We next analyze root causes of problematic failure handling using all SCGFailure instances observed in both **D1** and **D2**.

1) Method: Fig. 4 shows the results for three types of problematic failure handling, as well as two types of anticipated failure handling. We identify the problems at all three phases: ❶ detection, ❷ reporting and ❸ recovery.

Ideally, SCGFailures should be handled as follows. When an SCGFailure truly happens, this RLF should be quickly and correctly detected (true RLF detected), and immediately reported to the network for a prompt recovery (by piggybacking the measurement reports of candidate SCG cells). It should be recovered by proper RLs in presence of qualified SCG cells with acceptable radio quality and performance; Otherwise, if all candidate SCG cells are not acceptable, it should end with no SCG recovery.

Fig. 4 plots a finite state machine (FSM) based on the outcomes at each phase of all the SCGFailure instances. We use three key signaling messages (SCGFailureInformation, RRCReconfiguration and SCG Addition) which are used at the reporting and recovery phases. We extract the failure type (say, the RLF triggering event) reported in SCGFailureInformation to analyze detection outcomes. At the detection phase (❶), there are two possible states: false RLF detected (**F**) and true RLF detected (**T**). At the reporting phase (❷), the measurement reports of candidate cells might be piggybacked (**P**) or not piggybacked (**NP**). In the **P** branch, the reported cells might be qualified (**Q**) or not qualified (**NQ**). If there exists at least one qualified cell, the recovery procedure (❸) skips the subsequent configuration and measurement steps, and directly uses the SCG Addition command to add new SCG cells; This ends with SCG recovery if success. In all other cases (in the **NP** or **P+NQ** branch), the recovery phase (❸) starts with RRCReconfiguration to run a complete 4-step procedure to add qualified SCG cells. Through analyzing the signaling messages in all the SCGFailure instances, we find that no qualified cells will be reported as long as there are no piggybacked reports at the reporting phase (in the **NP** branch). In the **P+NQ** branch, RRC Reconfiguration Procedure is performed as anticipated: cells will be measured and reported as configured in RRCReconfiguration. Specifically, the device reports the measurement reports of candidate cells if their RSRP/RSRQ is stronger than the given RSRP/RSRQ threshold (B1 event [3]). It ends with two anticipated failure handling: no SCG recovery (without qualified SCG cells) and SCG recovery (with qualified cells).

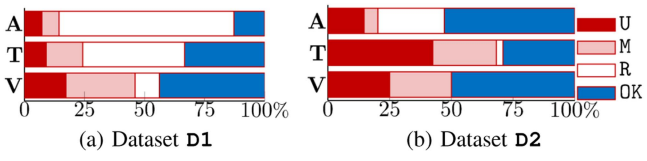


Fig. 5. Breakdown per SCGFailure type in **D1** and **D2**.

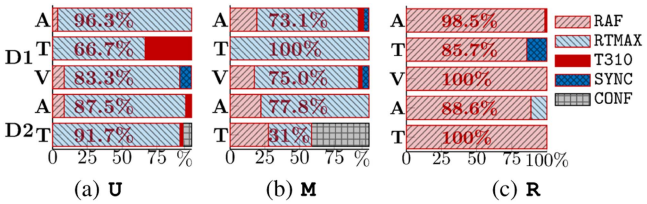


Fig. 6. Breakdown per triggering event.

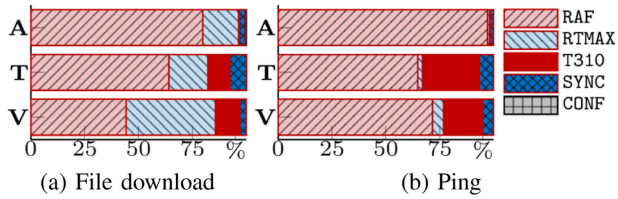


Fig. 7. Breakdown per SCGFailure type with two different applications (file download, ping) in **D1**.

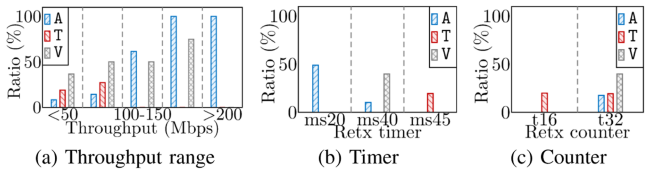


Fig. 8. The ratio of RTMAX events with different throughput ranges and parameter settings (timer and counter for RTMAX events) in **D1**.

Finally, we find that problematic failure handling comes in three forms: *unnecessary handling* (**U**), *missed recovery* (**M**) and *repeated failures* (**R**). **U** and **M** share the same problem of no recovery in presence of qualified cells. That is, we share the error path **NP+NQ** in the left of Fig. 4. Their difference is that **U** starts with a false RLF while **M** starts with a true RLF. **R** occurs when the newly added SCG cell suffers with the same failure which results in failure recovery.

2) Instance Breakdown: Before we dive into the root causes of problematic failure handling, we present the breakdown of all SCGFailure instances per type in Fig. 5. We have three observations.

First, problematic SCGFailure handling is quite common out of all failure instances: We notice that SCGFailures are not common (3.3% ≈ 4,275/129.8K in **D1**); However, once a SCGFailure occurs, it is likely handled in a problematic manner; For all three US operators, their ratios of normal failure handling all are below 50% in both **D1** and **D2**. *Second, the breakdown does vary with operators and test regions*. Operator **V** seems to do a better job while **A** and **T** suffer with more problematic SCGFailure handling in our study. The breakdown differs in **D1** and **D2** because 5G experiments are conducted in different cities. We admit that **D2** might be more biased as we intend to run more experiments at several locations of our interests. *Third, problematic failure handling also varies at locations.* In **D1**, 72.8% and 42.4% of SCGFailure instances are repeated failures with **A** and **T**. However, we notice that most instances with repeated failures take place in one or two small regions rather than evenly scatter at many locations. In contrast, unnecessary handling (**U**) and missed recovery (**M**) are observed at more places.

3) Root Causes: We next reveal how problematic failure handling occurs, namely, these key state transitions shown in Fig. 4.

○ *Unnecessary SCGFailure handling:* First, we find that RTMAX is the only dominant trigger to unnecessary failure handling. This matches with our illustrative instances (Section III-A). Fig. 6(a) shows the breakdown of unnecessary SCGFailure handling per trigger event. We skip **V** in **D2** because we do not see sufficient SCGFailure instances. RTMAX contributes to 83%–96% of **U** instances for all three operators in both datasets.

Unfortunately, more false alarms occur with heavier data traffic and higher throughput. We observe that RTMAX event is more likely to trigger false SCGFailures for heavy traffic users, especially when the SCG is offering very high throughput. To investigate the impact of user traffic on SCGFailure triggering, we separate dataset **D1** based on the application type (file download and ping), and compare the proportion of unnecessary SCGFailures for these applications. As depicted in Fig. 7, SCGFailures triggered by RTMAX event are rarely observed under light traffic (ping), but they are not uncommon under heavy traffic (file download). When the application switches from ping to file download, the ratio of SCGFailures triggered by RTMAX event significantly increases from only 1%–5% to 18%–41%. Correspondingly, the proportions of RAF and T310 events decrease under the file download application. Next, we further categorize SCGFailure instances with file download application into five groups based on the throughput before the failure, ranging from <50 Mbps to >200 Mbps. Fig. 8(a) presents the ratio of SCGFailures triggered by RTMAX event in each throughput range for all three operators in **D1**. For both **A** and **V**, we observe a clear trend that with higher throughput, RTMAX event is more likely to occur. We do not have SCGFailure instances with >100 Mbps throughput for **T**, so we cannot determine the trend for this operator. For **A**, with <50 Mbps throughput, RTMAX event only accounts for less than10% of SCGFailures. However, when the throughput exceeds >150 Mbps, all SCGFailures are triggered by RTMAX event. Similarly, for **V**, as the throughput increases from <50 Mbps to > 150 Mbps, the ratio of RTMAX events doubles, rising from 37% to 75%.

Last but not least, aggressive parameter setting greatly increases the likelihood of RTMAX events. As introduced in Section II-A, the triggering of RTMAX event is decided by two key parameters: the retransmission timer `t-PollRetransmit` and the counter `rlc-MaxNumRetx`. We find that the setting of the `t-PollRetransmit` timer significantly affects the triggering of RTMAX event, while the `rlc-MaxNumRetx` counter has almost no impact. Fig. 8(b) plots the ratio of RTMAX events with different `t-PollRetransmit` timer settings of all three operators. We see that **A** adopts two timer settings, 20 ms and

40 ms, while **T** and **V** use only one timer setting each, 45 ms for **T** and 40 ms for **V**. When **A** uses the 20 ms timer, RTMAX events are much more likely to occur compared to the 40 ms timer. The ratio of RTMAX events soars from only 10% with the 40 ms timer to 49% with the 20 ms timer. Operators may also use different settings for the rlc-MaxNumRetx counter. For example, Fig. 8(c) shows that **T** sets 16 or 32 as the threshold of counter rlc-MaxNumRetx. However, we observe that this counter setting does not significantly impact the likelihood of triggering RTMAX events. For **T**, the proportion of RTMAX events remains around 20%, regardless of whether the counter is set to 16 or 32. Therefore, the t-PollRetransmit timer setting is the key factor impacting the triggering of RTMAX events. When the timer is set to a very low value (e.g., 20 ms), the difficulty of retransmission increases sharply, making RTMAX events more likely to be triggered.

Takeaway: The triggering mechanism of SCGFailures must consider two critical factors: user traffic load and the setting of the retransmission timer t-PollRetransmit. Otherwise, the mechanism cannot distinguish whether the RTMAX event is caused by a broken radio link, excessive traffic load, or an overly aggressive timer setting. In the latter two cases, SCGFailures are very likely to be falsely triggered. Therefore, it is essential to design a novel SCGFailure detection mechanism to address these issues.

○ *Repeated SCGFailures:* We next investigate how repeated failures are triggered. We see that RAF is the dominating event for repeated failures. As shown in Fig. 6(c), 85.7%-100% of repeated failures are triggered by RAF for all operators. RAF is designated to release the poor SCG when the device fails to complete random access to this SCG cell. It often happens when the RSRP of this SCG cell is below a certain threshold. However, the operator might set the RSRP threshold for SCG Addition below the threshold needed for random access. More precisely, when the actual RSRP is larger than the threshold for SCG Addition (but smaller than the threshold for random access ($\text{RSRP}_{SCGAddition} < \text{RSRP} < \text{RSRP}_{RA}$), the device re-connects to the failed cell and the SCGFailure is persistently repeated.

Unfortunately, the problem of mismatched RSRP thresholds cannot be resolved by simply setting a fixed higher RSRP threshold for SCG addition. This is because the RSRP threshold required for random access varies significantly across different regions, ranging from −110 dBm to −80 dBm. For instance, in a specific region R4 (**A**), the RSRP threshold needed for random access is much higher than in other regions of **D1**. This greatly increases the difficulty of random access, resulting in a remarkably higher ratio of repeated failures in R4. Fig. 9 illustrates an example of repeated failures in R4. In this example, we repeatedly drove over a same route, causing the RSRP to fluctuate periodically within the range of −100 dBm to −60 dBm. We observe that once the RSRP drops to −80 dBm or lower, repeated random access failures occur. When the RSRP exceeds −70 dBm, the repeated failures cease, and the UE can successfully access and use the 5G cells. Such instances are very common in R4, whereas in other regions, an RSRP of −80 dBm almost guarantees that random access failures will never happen. Fig. 10 plots the RSRP threshold needed for random access in each region in **D1**. For each region, we calculate the ratio of normal handovers with RSRP higher than the threshold, and the ratio of random access failures with RSRP lower than the threshold. We select the threshold that maximizes the sum of these two ratios to effectively distinguish random access failure instances and normal handovers. We see that the RSRP thresholds in R3 and R4 are around −80 dBm, which is at least 20 dB higher than the thresholds in all other regions. Consequently, random access failures frequently happen in R3 and R4, accounting for 95%-99% of all SCGFailure instances.

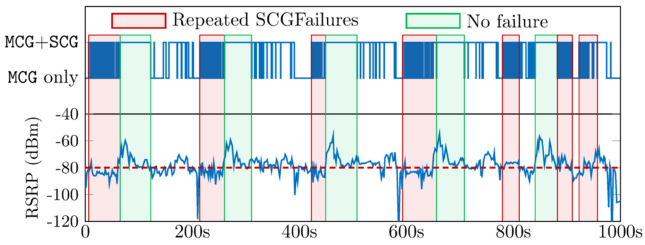


Fig. 9. An example of repeated failures in R4 (**A**) in **D1**.

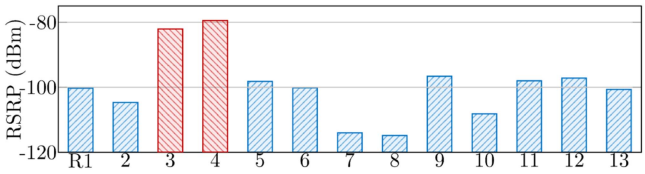
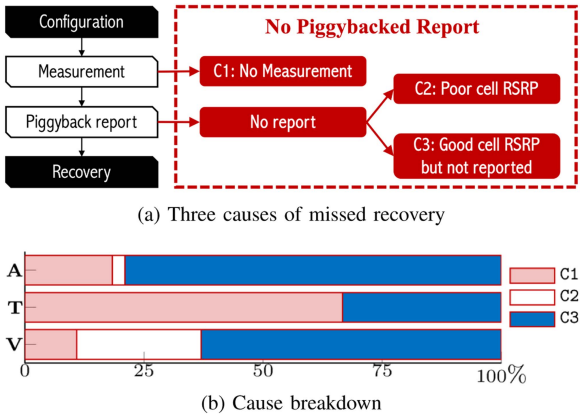


Fig. 10. RSRP threshold for random access in each region in **D1**.

Takeaway: The same RSRP level could have completely different meanings in different regions, and no single RSRP threshold can work universally. The RSRP threshold in SCG recovery should be adaptive to region-specific RSRP thresholds for random access to avoid repeated failures.

○ *Missed SCGFailure recovery:* We finally introduce how missed recovery failures happen. For missed recovery failures, the dominant triggering event (RTMAX) is same to unnecessary failures. Fig. 4 shows that the only difference between unnecessary failure handling (**U**) and missed recovery (**M**) is that a **M** instance is triggered correctly with a true RLF, while **U** with a false one. When a SCGFailure is correctly triggered but without piggybacked measurement reports of neighboring SCG cells, no qualified SCG cells will be reported; Consequently, MCG cannot send the SCG Addition command to the UE without candidate SCG cell information, and the UE thus misses the chance of recovery to good SCG cell(s).

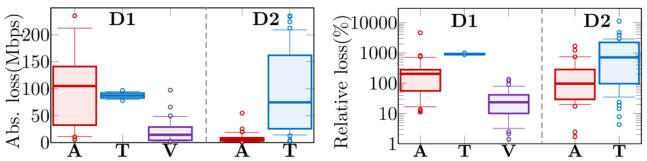
We next delve into the root causes of missing piggybacked measurement reports. As illustrated in Fig. 11(a), sending the SCG Addition command with a piggybacked measurement report involves three essential steps: configuration, measurement and piggybacking the report. The piggybacked report could be missed due to three different causes during the measurement and piggybacking report steps: **C1**: No candidate cells are measured; **C2**: Candidate cells are measured but their RSRP is lower than the report threshold; **C3**: Candidate cells with qualified RSRP are measured but not reported. Fig. 11(b) shows the breakdown of causes for all missed recovery failure instances in **D1**. We find that in most missed recovery instances for **A** and **V**, the



(a) Three causes of missed recovery

(b) Cause breakdown

Fig. 11. Causes of missed SCGFailure recovery.



(a) Absolute loss

(b) Relative loss

Fig. 12. Absolute and relative throughput loss of unnecessary SCGFailure handling (**U**).

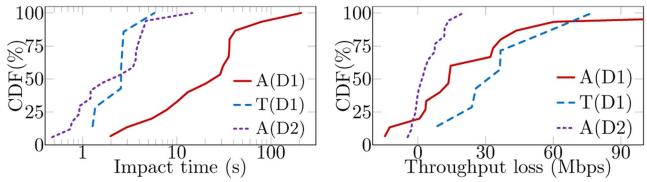
missing of piggybacked measurement report is not due to the lack of cell measurement or poor radio quality. For **A** and **V**, good candidate cells have already been measured before SCGFailure being triggered in 63% and 79% of instances. However, these cells are not piggybacked in the `SCG Addition` command, so the network side is unaware of these qualified cells and cannot guide UE to select a target cell for SCG recovery.

Takeaway: Device-side cell measurement information can be leveraged to address the problem of missed failure recovery. When the measurement report is not piggybacked, network lacks essential cell information to guide UE. However, by accessing recent measurement results on the device side, UE can identify suitable cells for SCG recovery by itself. Therefore, it is feasible to design a device-side solution to address the missed recovery problem. We will elaborate how we implement this idea in our solution in Section IV-A.

C. Performance Impacts

We next present negative performance impacts of problematic SCGFailure handling per type. Improper failure handling results in substantial throughput loss; We observe that download speed drops by half or more in most instances; It even declines by one order of magnitude (up to two orders of magnitude) in a few instances.

*Unnecessary SCGFailure handling (**U**):* We define two metrics to assess the resulting throughput loss – (1) *absolute loss*: the gap of average throughputs in 10 seconds before and after a SCGFailure; (2) *relative loss*: the ratio between absolute loss and throughputs after a SCGFailure. Fig. 12 plots the distributions of the absolute and relative throughput loss with three US



(a) Impact time

(b) Thput. loss

Fig. 13. Negative impacts of repeated SCGFailures (**R**).

operators; **V** in **D2** is skipped without sufficient instances. We have two observations.

First, throughput loss greatly varies with operators: In terms of relative loss, **T** suffers more throughput degradation than **A** and **V**. For **T**, download speed drops by more than one order of magnitude in almost all instances in **D1** and 41.7% of instances in **D2**; The worst instance was observed in **D2**, with a 111.5-fold decline from 142.2 Mbps to 1.3 Mbps (median). For **A**, download speed declines by more than half (namely, the relative loss $>$ 100%) in 63% of instances in **D1** and 50% of instances in **D2**. Compared to **T** and **A**, **V** does the best job with its median loss below 30%.

Second, throughput impacts are largely consistent in terms of absolute and relative loss and inconsistent patterns are caused by various data speed before failures occur: Interestingly, we see that **A** has distinct patterns in terms of the absolute and relative loss in both datasets. Although its relative loss is similar in **D1** and **D2**, the absolute loss is much lower in **D2**. For **A**, the absolute loss is $>$ 100Mbps in 51.9% of instances in **D1**, but $<$ 10 Mbps in 79.2% of instances in **D2**. Specifically in **D1**, **A** has the median loss of 105 Mbps (25th/75th percentile: 32.4 Mbps/141 Mbps), which is even higher than 87.3 Mbps with **T**. We further examine why. It turns out that such distinct impacts are caused by various 5G deployment. **A** deploys mmWave cells with much larger bandwidth (100 MHz) in **D1** but uses narrow channels over Sub-6GHz (10 MHz) in **D2**. The use of mmWave cells allows much higher throughput than 5G over Sub-6GHz. With much higher data throughout prior to SCGFailures, **A** thus loses much more absolute speed in **D1**; In **D2**, although the absolute loss is much smaller, negative impacts are not negligible; Data speed still reduces by half in more than 50% of instances. In contrast, **T** deploys 5G cells on the same sub-6GHz band (n41) in both datasets and the resulting impacts are consistent in these two datasets. Compared to **A**, **T** achieves higher data speed over sub-6GHz because it uses wider channels (bandwidth: 60/100MHz).

*Missed recovery SCGFailure handling (**M**):* We observe huge performance impacts when SCGFailure recovery is missed. To assess performance impacts of each **M** instance, we compare data throughput in two scenarios: (1) the reality without SCG cells being recovered, and (2) a what-if case with active SCG cells on the same location. We calculate the absolute and relative throughput loss between (1) the average throughput in a short time period (10 seconds) just after the SCGFailure occurs and (2) the median throughput with active SCG cells. Fig. 14 plots the results.

In terms of absolute throughput loss, **T** performs worse than **A** (and **V**). **T** loses more than 100 Mbps in 40% (**D1**) and 65.5%

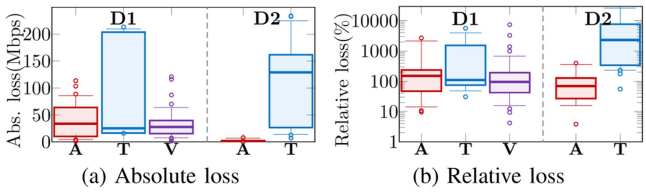


Fig. 14. Absolute and relative throughput loss of missed SCGFailure recovery (**M**).

(**D2**) of instances, even with the absolute throughput loss up to 326 Mbps (**D2**). In contrast, **A** loses much less than **T**, with its absolute loss below 70 Mbps in most instances; Note that **A** experiences distinct throughput loss in these two datasets: the median throughput loss is below 40 Mbps in **D1** and even below 5 Mbps (1.6 Mbps) in **D2**. This is also caused by various 5G deployment as explained above.

It is worth noting that the median throughput loss of unnecessary failure handling (**U**) is much larger than **M** in **D1**; For **A**, it declines from 105 Mbps to 34 Mbps; For **T**, the loss of missed recovery is more diverse but its median throughput loss also decreases from 87 Mbps to 26 Mbps. It implies that **U** poses more negative impacts than **M**, in terms of absolute throughput loss for **A** and **T** in **D1**.

In term of relative loss, data throughput reduces more than by half (say, relative loss = 100 %) in more than 50% instances in **D1** (**A**: 57.7%, **T**: 60%, **V**: 50%). For **T** in **D2**, download speed declines by more than one order of magnitude in 96.6% of instances. We note that the relative loss due to missed recovery (**M**) is higher the one due to unnecessary handling (**U**), which is different from the conclusion in terms of absolute throughput loss. It is because that absolute data speed without failure recovery is smaller in **D2**. It is not hard to understand; **D2** is collected in West Lafayette, a much smaller city. Compared to **D1**, both **A** and **T** offer lower data speed in **D2**, regardless of the use of 5G.

Repeated SCGFailure handling (R): In our study, most repeated SCGFailures are observed in three settings: **A** (**D1**), **T** (**D1**) and **A** (**D2**), as shown in Fig. 5. We thus use them to assess performance impacts of repeated failures (**R**). Fig. 13 plots the cumulative distribution functions (CDFs) of impact time and throughput loss. In every **R** instance, we use the interval from the first failure to the last one as the impact time, which is actually a lower bound of the actual impact time; The throughput loss is calculated as the absolute gap between the average throughput during the impact time and the median throughput without SCGFailures at the same location. We have two observations.

First, the impact time lasts much longer in **A** (**D1**) than **A** (**D2**) and **T** (**D1**). In **A** (**D1**), repeated failures last more than 30s in 40% of instances and the duration even goes up to >200s. In **A** (**D2**) and **T** (**D1**), most repeated failures are shorter than 5s. Second, **A** (**D2**) has the minimal throughput loss, which is somehow consistent to those observed in the **U** and **M** instances. It is because **A** offers low data speed even without failures in **D2**. In terms of throughput loss, **T** (**D1**) is worse than **A** (**D1**), despite shorter impact time. In **D1**, **T** loses more than 30 Mbps in 50% of instances and **A** in 33% of instances.

IV. SFM: Solution & Evaluation

Inspired by our findings, we propose SCGFailure Manager (SFM) which is the first solution to systematically solve problematic SCGFailure handling based on our knowledge.

A. The Design of SFM

Before introducing the design of SFM, an important question is to be answered: where is the best location to deploy SFM? There are three options: (1) device-side, (2) network-side, or (3) hybrid, which means that some modules are deployed on the device and the rest on the network side. Our decision is to design SFM as a solution that is entirely deployed on the device side, based on the following reasons: First, in the detection of SCGFailure, regardless of the triggering event of SCGFailure, UE is always the first to be aware of it and it has full knowledge (data performance, signaling messages, radio measurement results) to determine whether the SCGFailure is correctly triggered and how to recover the SCG. Second, when channel quality of both MCG and SCG is poor, UE may not be able to perform normal signaling exchange with the network. In such cases, a device-side solution can help UE perform self-recovery without waiting for instructions from the network to resolve SCGFailure.

In our measurement study in Section III, we have obtained many critical insights, which are utilized by SFM as domain knowledge to resolve problematic SCGFailure handling. Here are the basic ideas how SFM leverages these insights to identify and address each type of problematic SCGFailure handling: (1) First, unnecessary failure handling (**U**) is mainly caused by improper triggering of RTMAX event based on the insights of our measurement study. When RTMAX event is triggered, if UE can still enjoy good data performance and there is no significant throughput degradation, the detected SCGFailure is identified as a false alarm and the SCGFailure reporting will be stopped by SFM. (2) Second, missed failure recovery (**M**) happens when there is no piggybacked measurement report of neighboring cells in SCGFailure report. Therefore, any SCGFailure instances without piggybacked measurement report are identified as **M**, and SFM will utilize the information stored on the device side to instruct UE to autonomously conduct SCG recovery. (3) Finally, repeated failures (**R**) are identified when SFM observes continuous random access failures. To interrupt the loop of SCG addition and SCGFailures, the threshold of SCG addition is gradually increased after each random access failure, so that it can converge to a proper level to add normal SCG and exclude risky SCG of random access failure.

Fig. 15 depicts the main flow of SFM, which comprises three critical modules *traffic-aware failure detection*, *device-side target cell selection* and *adaptive threshold adjustment*. These three modules address **U**, **M**, and **R** respectively.

○ *Traffic-aware failure detection:* (Algorithm 1) In the detection stage, SFM performs traffic-aware failure detection to avoid false alarms of SCGFailure. When an RTMAX event is triggered, SFM utilizes recent traffic information on the device side to verify whether it indicates a genuine radio link failure. This verification involves two steps. First, SFM conducts a pre-check on the traffic type and the retransmission timer setting (line 7

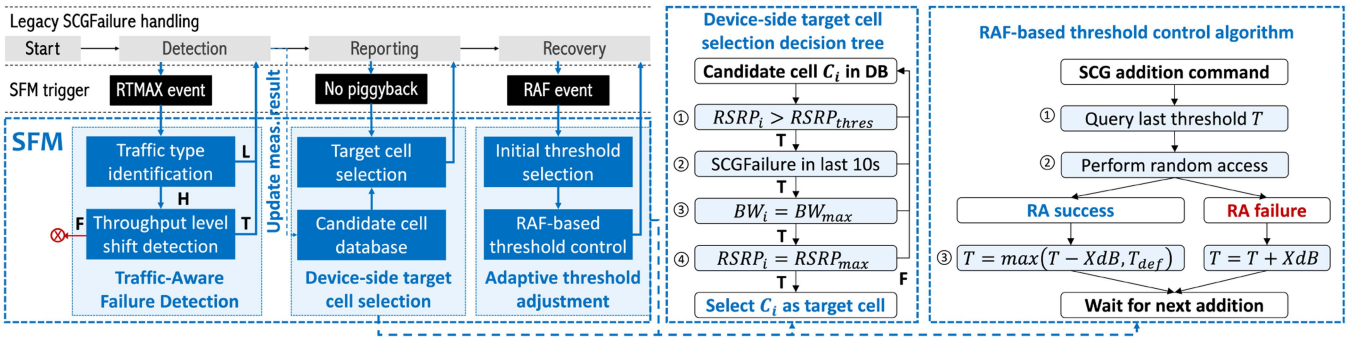


Fig. 15. The design of our solution SFM.

Algorithm 1: Traffic-Aware Failure Detection.

Input: $event$, $timer_{rt}$, $X = \{x_{t_1}, x_{t_2}, \ldots, x_{t_n}\}$
// Trigger event, retransmission timer and throughput samples at each timestamp t_i

// Calculate average throughput in recent 10s
$X_{10s}^{\text{avg}} \leftarrow \sum X_{10s}/\text{len}(X_{10s})$
// For RTMAX *event on heavy traffic user with low timer*
if $event =$ RTMAX **and** $X_{10s}^{\text{avg}} > th_{heavy}$ **and** $timer_{rt} < th_{timer}$ **then**
    // Use CUSUM to detect throughput drop count in recent 10s and 1min
    $thput\_drop\_10s \leftarrow CUSUM(X_{10s}, k, h)$
    $thput\_drop\_1min \leftarrow CUSUM(X_{1min}, k, h)$
    if $thput\_drop\_10s = 0$ **then**
        // No throughput drop in recent 10s
        Don't report SCGFailure
    else if $thput\_drop\_1min > \alpha * thput\_drop\_10s$ **then**
        // Throughput drops are consistently observed
        Don't report SCGFailure
    else
        Report SCGFailure
    end if
else
    Report SCGFailure
end if

of Algorithm 1). It is because the falsely triggered RTMAX events are accompanied by heavy traffic and aggressive parameter setting, as illustrated in Section III-B. SFM determines whether the user is experiencing heavy or light traffic based on their average throughput X_{10s}^{avg} in the 10 seconds before SCGFailure. If the average throughput exceeds a set threshold th_{heavy}, the user is considered to be using heavy traffic. Additionally, SFM checks whether the retransmission timer is set to a low value. By default, SFM sets the throughput threshold th_{heavy} at 1 Mbps and the retransmission timer threshold th_{timer} at 40 ms. If conditions meet these thresholds, SFM performs a further inspection.

Second, for heavy traffic users, SFM detects throughput degradation before SCGFailure. If there is a significant drop in throughput, it likely indicates a broken radio link causing continuous data retransmissions. In such cases, it is necessary to report SCGFailure to switch to a new SCG to recover performance. Conversely, if throughput remains consistently high, an RTMAX event does not imply poor SCG performance, and thus, SCGFailure should not be reported. To detect throughput degradation, SFM uses Cumulative Sum (CUSUM) algorithm, a widely used statistical technique for change detection. CUSUM can effectively detect shifts in the mean level of a monitored metric (here, DL/UL throughput). The reason we choose the classic CUSUM algorithm instead of a machine learning-based approach is that machine learning methods typically require pre-training and large datasets to achieve good performance, and they also consume more computational resources. Therefore, classic algorithms like CUSUM are better options for real-time solutions deployed on mobile devices. As shown in line 9 and line 10, CUSUM has two key parameters: sensitivity parameter k and decision threshold h. We test the detection performance of SFM under different parameter settings (Section IV-B) and select $k = 5$ and $h = 20$ as the default settings. For each triggered RTMAX event, SFM applies the CUSUM algorithm to detect any level shift in throughput within 10 seconds before the RTMAX event. If no throughput drop is detected, SFM will discard SCGFailureInformation to stop the reporting of SCGFailure, and the UE will continue to use the current SCG (line 11-13).

The current detection method still has an important remaining issue. Apart from radio link failures, throughput drop may be caused by other reasons, such as dynamic radio condition (e.g., on high speed railway) or the input from application-layer. In these cases, the throughput drop may mislead the decision of SFM and cause unnecessary SCGFailures. To handle these cases, SFM checks recent traffic information again and records the number of throughput drops in the last minute before SCGFailure request. If the count of throughput drops is higher than α (default value: $\alpha = 5$) times of the throughput drop number in the recent 10s (line 14-16), SFM considers that the throughput drops are persistently observed and they are very likely to be caused by other reasons rather than radio link failure, and SFM will stop the reporting of SCGFailure. Through this approach, SFM can be generalized to more scenarios.

◦ *Device-side target cell selection:* (Algorithm 2) Next, in the reporting stage, SFM offers device-side target cell selection, when reporting from UE to network side is blocked. When the measurement report is not piggybacked in SCGFailureInformation, SFM instructs UE to stop waiting for commands

Algorithm 2: Device-Side Target Cell Selection.

1: **Input:** $DB = \{c_i : (bw_i, rsrp_i, t_i^{\text{fail}})\}$
2: *// The cell database with bandwidth, RSRP and last SCGFailure timestamp of each candidate cell* c_i
3:
4: *// Check the maximum bandwidth and RSRP*
5: $bw_{\max} \leftarrow \max(bw_i)$ for all c_i in DB
6: $rsrp_{\max} \leftarrow \max(rsrp_i)$ for all c_i in DB
7: *// Trigger* SFM *when no piggybacked report*
8: **if** no piggybacked measurement report in `SCGFailureInformation` **then**
9: *// Check each candidate cell in database*
10: **for** c_i in DB **do**
11: *// Filter poor candidate cells*
12: **if** $rsrp_i > th_{rsrp}$ **and** t_i^{fail} is in recent 10s **then**
13: *// Select the best candidate cell*
14: **if** $bw_i = bw_{\max}$ **and** $rsrp_i = rsrp_{\max}$ **then**
15: **Select** c_i **as target cell of SCGrecovery**
16: **end if**
17: **end if**
18: **end for**
19: **end if**

Algorithm 3: Adaptive Threshold Adjustment.

1: **Input:** Last threshold T, adjustment factor X, default threshold T_{def}
2: **When UE receives `SCG` addition command**
3: Query the last threshold T
4: Perform random access (RA)
5: **if** RA success **then**
6: $T \leftarrow \max(T - XdB, T_{def})$
7: **else**
8: $T \leftarrow T + XdB$
9: **end if**
10: Wait for the next `SCG` addition

from network; Instead, SFM uses the measurement information stored on the device to select an appropriate target cell for `SCG` recovery.

To achieve this, SFM first constructs and maintains a candidate cell database on the device side. For each candidate cell recently measured (e.g., within the last 10 seconds), the database records the following key information: (1) Basic cell information, including cell ID and bandwidth, (2) Most recent RSRP measurement result of the cell, (3) The timestamp of the last SCGFailure that occurred on this cell. When `SCGFailureInformation` is sent without piggybacking measurement reports (`line 8` of Algorithm 2), SFM initiates device-side cell selection based on the historical data in the cell database. As shown in Fig. 15, SFM uses a four-level decision tree to decide the target cell for `SCG` recovery: For each recently measured cell, SFM first determines whether it is a valid option for `SCG` recovery. At the first and second layer, SFM filters out cells with RSRP below the threshold th_{rsrp} and cells that have recently experienced SCGFailures (`line 12`). Next, SFM selects the cell with the best overall performance from the qualified candidates (`line 14`). It considers both radio strength and bandwidth resources of candidate cells, two critical factors deciding the cell performance [10]. At the third layer, SFM chooses all cells on the channel with the maximum bandwidth; Among these cells, SFM selects the one with the highest RSRP as the final target cell.

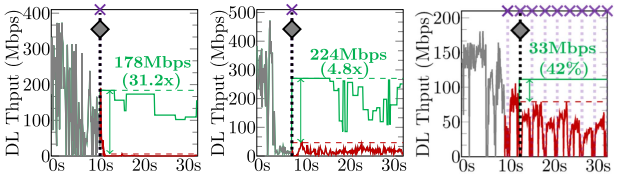
○ *Adaptive threshold adjustment:* (Algorithm 3) In the final recovery phase, SFM dynamically adjusts the RSRP threshold for `SCG` addition to avoid repeated random access failures. Fig. 15 illustrates our threshold control algorithm. When UE receives an `SCG` addition command, SFM first checks the RSRP threshold T from the last successful `SCG` addition (`line 3` of Algorithm 3). SFM then adjusts the threshold based on the outcome of the random access attempt. If the random access fails, the threshold T is increased by X dB (`line 8`); if the random access succeeds, the threshold T is decreased by X dB, but not below the default threshold T_{def} configured by the operator (`line 6`). The default value for X is set to 5 dB to balance efficiency and accuracy of threshold control based on our tests. UE uses the new threshold to determine whether to proceed with the next `SCG` addition. Finally, if random access failures are not observed for more than 1 h, threshold T is reset to default value T_{def}.

Under the control of our algorithm, when random access failures occur, the rapid increase of RSRP threshold will prevent subsequent `SCG` additions and repeated failures. Although the UE temporarily loses the `SCG` connection, the `MCG` connection can ensure the stability of cellular service. After a successful random access or a long period without observing failures, SFM will lower the threshold to probe for the lowest feasible threshold.

B. Evaluation

The deployment of SFM in operational cellular networks requires operators to modify the logic of detection and reporting SCGFailure on the UE side. Besides, operators need to modify the policy of cell selection on the network side to implement the module of adaptive threshold adjustment. Since we are unable to get the permission from operators to deploy SFM in operational 5G networks, we adopt a trace-driven approach to validate and evaluate its effectiveness. We conduct a "what-if" study in two steps. First, we run SFM on each collected SCGFailure instance to determine: (1) changing `SCG` commands, such as cancel SCGFailure reporting or prevent `SCG` addition, (2) cells used for `SCG` after the SCGFailure instance. For the new `SCG` selected by SFM, there are three possible cases: (1) if SFM determines that the SCGFailure should not be triggered, the new `SCG` remains the same as the original `SCG` before the SCGFailure, (2) when there is no piggybacked measurement report, SFM selects the new `SCG` based on previous measurement results, (3) if the RSRP threshold is raised due to random access failure and `SCG` addition is prevented, the new `SCG` will be empty and the UE will use `MCG` only. Second, when SFM selects a different `SCG`, we estimate the performance of new `SCG` by checking its historical throughput at the current location. We then compare the performance of the new `SCG` with that under legacy handling to assess the impact of SFM on overall performance.

We use the three examples in Fig. 1 to further explain how SFM works under three different problematic SCG failure handling scenarios, and how we evaluate its performance gains.



(a) Unnecessary (**U**) (b) Missed recovery(**M**) (c) Repeated (**R**)

Fig. 16. Three instances illustrate how SFM works on three types of "problematic" SCGFailure handling (**U**, **M**, **R**) and improves throughput (♢: SCG changed by SFM).

Fig. 16 illustrates when SFM makes a decision to select a different SCG from the legacy handling (marked by ♢) in these three examples, and compares the throughput of the new SCG with the legacy SCG. In the first example (unnecessary SCGFailure handling), since there is no significant degradation in throughput, SFM prevents the reporting of SCGFailure, and UE continues to use the original SCG. With SFM, the UE maintains around 180 Mbps throughput, significantly higher than the less than 10 Mbps throughput under legacy handling. In the second instance of missed recovery failure, SFM detects the missing of piggybacked measurement report in the SCGFailureInformation and autonomously selects a 100 MHz mid-band cell from the recently measured cells as the new SCG. Under legacy handling, the UE is served by MCG only after SCGFailure, resulting in a throughput of around 30 Mbps; With the new SCG selected by SFM, UE boosts the throughput to 250 Mbps. The third example shows how SFM handles repeated failures. After the first random access failure at 9.5s, SFM increases the RSRP threshold for SFM addition from −82 dBm to −77 dBm. This successfully prevents UE from attempting to reconnect to the failed cell at 12.1s, keeping UE in the MCG only state. Compared to the highly fluctuating throughput with legacy failure handling, SFM improves the overall throughput by at least 40%.

Next, we assess the performance benefits of SFM on two applications: file downloading and file uploading.

File downloading: We first examine whether SFM can effectively prevent problematic SCGFailure handling in the application of file downloading. Fig. 17(a), (b), and (c) show the proportion of three types of problematic SCGFailure handling (**U**, **M**, **R**) under the control of SFM and legacy mechanism. Except for missed recovery failures (**M**) in **T**, SFM resolves 60%-80% of instances for each type of problematic SCGFailure handling. For unnecessary SCGFailures (Fig. 17(a)) and missed recovery SCGFailures (Fig. 17(b)), SFM reduces at least two-thirds of problematic instances for **A** and **V**. The proportion of problematic instances significantly decreases from 7%–29% to 1%–8%.

For **T** users, SFM reduces the proportion of unnecessary SCGFailures from 5%–42% to 1%–12%, though its effect on missed recovery failures is less impressive. In **D1** and **D2**, SFM reduces the proportion of **M** instances from 15% and 26% to 12% and 17% respectively, so the reduced ratios are only 20% and 35%. This is because to fix missed recovery failures, SFM requires that candidate cells have been measured by UE before the failure. However, according to Fig. 11(b), candidate cells are measured in only about 30% of **T** instances. Consequently, there is no optimization room for SFM in these instances with no measurement, explaining why it can only fix a small portion of **T**'s instances without SCG recovery.

SFM shows excellent performance in terminating repeated failures in advance. Fig. 17(c) indicates that for **A** and **T**, SFM avoids more than 75% of repeated failures across all three datasets. The proportion of repeated failures decreases from up to 76% (**T** in **D3**) to no more than 12%. We further compare the lengths of repeated failures under SFM and legacy handling in Fig. 17(d), showing how many consecutive SCGFailures are in each series of repeated failures. SFM typically terminates repeated failures within the first two failures. Even in the worst case, the number of repeated failures with SFM does not exceed 15, while under legacy handling, repeated failures can occur more than 50 times consecutively.

Finally, we test the impact of parameter settings on detecting problematic SCGFailure handling. As mentioned in Section IV-A, in the traffic-aware failure detection module of SFM, the setting of parameter k and h can significantly influence the detection results. If k and h are set too high, SFM will miss the detection of many unnecessary SCGFailures; if k and h are set too low, some necessary SCGFailures (including **M**, **R** and normal SCGFailures) may be falsely detected and canceled by SFM. Fig. 19(a) and (b) show the curves of the missed detection ratio of unnecessary SCGFailures and the false detection ratio of necessary SCGFailures in datasets **D1** and **D2** when we vary the setting of k and h. It can be seen that the setting $k = 5$ and $h = 20$ can effectively balance detection accuracy and recall. Under this seeting, SFM can detect 85%-95% of unnecessary SCGFailures while keeping the proportion of falsely canceled necessary SCG failures at around 10%. Other settings either fail to achieve high recall or result in a significant amount of false detections. Therefore, we select $k = 5$ and $h = 20$ as the default parameter setting.

Next, we quantify the downlink throughput gains achieved by SFM by fixing each type of problematic SCGFailure handling. For each SCGFailure instance, we calculate two metrics Δ and γ to evaluate the absolute and relative throughput gains of SFM, respectively: $\Delta = TP_{\text{SFM}} - TP_{legacy}$, $\gamma = \Delta / TP_{legacy}$. Here, TP_{SFM} is the median throughput of the new SCG selected by SFM at the current location, and TP_{legacy} is the median throughput within 10 seconds after the SCGFailure under legacy handling.

Fig. 18 presents the CDF of throughput gains by SFM. First, SFM significantly improves throughput after SCGFailure, and it doubles the data throughput in 50%-75% of instances. For all three operators, the relative throughput gain γ is greater than 1 in 75% of repeated failure instances in all datasets (Fig. 18(f)). In other two cases (Fig. 18(b) and (d)), throughput is also boosted by at least 100% with SFM in half of instances. Second, among the three operators, throughput gains by SFM are most significant for **T**. In 30%-50% of instances, SFM can even increase throughput by at least 10 times. In Section II-I-C, we have revealed that problematic SCGFailure handling has the severest impact on **T**. SFM successfully translates this greater improvement potential into the largest throughput gains for **T**. Finally, SFM achieves the highest absolute throughput gain by fixing unnecessary SCGFailures. Fig. 18(a) shows that SFM improves throughput by at least 100 Mbps in half of the

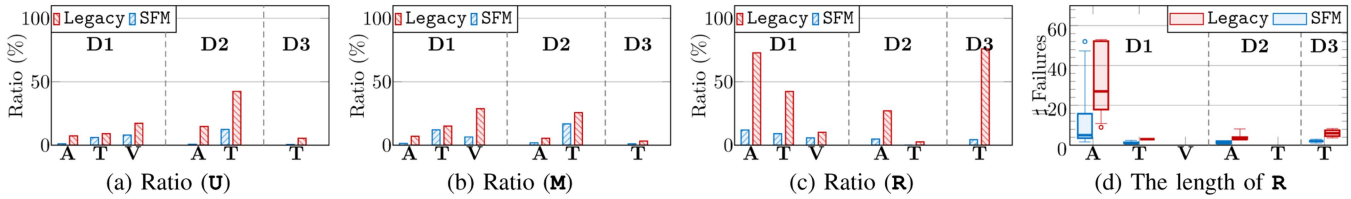


Fig. 17. The ratio of problematic SCGFailure handling (**U**, **M**, **R**) and the length of repeated failures (**R**) in file downloading.

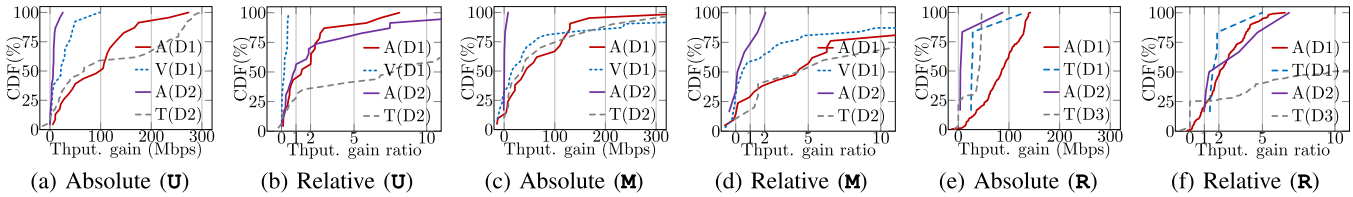


Fig. 18. CDF of the absolute and relative throughput gains with SFM in file downloading.

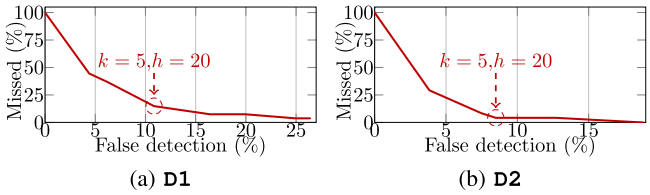


Fig. 19. The curve of missed **U** ratio and falsely canceled SCGFailure reporting ratio.

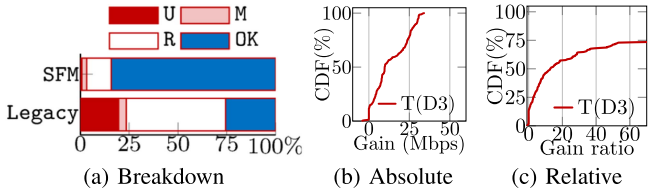


Fig. 20. SCGFailure type breakdown and throughput gains with SFM in file uploading.

instances for **A** in **D1** and **T** in **D2**. This is because unnecessary SCGFailure are typically triggered when the throughput is very high. Therefore, compared to the other two cases (**M**, **R**), SFM realizes higher absolute throughput gains in **U** instances.

File uploading: To evaluate the performance gain of SFM in bulky file uploading application, we conduct file uploading experiments while collecting dataset **D3**. Our first observation is that repeated failures are the dominant SCGFailure type for both file uploading and file downloading. Fig. 17(c) and 20(a) show that for both applications, more than half of SCGFailure instances in **D3** occur repeatedly within a short period. Additionally, in file uploading, unnecessary SCGFailure is the second most common type, accounting for 20% of SCGFailure instances. In contrast, the ratio of unnecessary SCGFailure is less than 5% in file downloading (Fig. 17(a)). This indicates that continuous retransmissions are more likely to be triggered during file uploading, leading to more false detections of SCGFailures.

Our evaluation results demonstrate that SFM can effectively address problematic SCGFailure handling and significantly improve uplink throughput in file uploading application. Fig. 20(a) shows the proportion of each SCGFailure type under SFM and legacy handling in the **D3** file uploading dataset. SFM resolves more than 70% of the problematic instances, significantly reducing their total ratio from 75% to 20%. Specifically, the proportion of repeated failures drops from 50% to only 15%, and SFM avoids almost all unnecessary failures. Fig. 20(b) and (c) show the absolute and relative gains in uplink throughput with SFM respectively. Due to the inherently lower ceiling of uplink throughput, the absolute gain of SFM in file uploading is lower than in file downloading. However, the relative gain in file uploading is remarkable. Fig. 20(c) shows that SFM increases uplink throughput by an order of magnitude in half of the instances. More impressively, in 30% of the instances, SFM even achieves over a 60-fold increase in uplink throughput.

V. Related Work

SCGFailure measurement: To the best of our knowledge, the preliminary version of this work [11] is the *first* measurement study to reveal problematic SCGFailure handling in reality. It was inspired by our recent work to examine misconfiguration in 5G networks as the number of serving cells advances from 1 to N [12]. Unlike previous studies on radio access failures on MCG [13], [14], [15], [16], [17], [18], our work focuses on SCGFailure handling, particularly when failure handling goes wrong. In this work, we substantially extend [11] by conducting an in-depth root cause analysis of all three types of problematic SCGFailure handling, providing valuable insights for designing effective solutions.

SCGFailure handling: Several studies [19], [20], [21], [22], [23], [24], [25], [26] in the literature have explored solutions to enhance SCGFailure handling. To reduce the probability of SCGFailures, [19] proposes an algorithm that adjust parameters such as T304 and T310 to maintain the failure rate at a pre-set threshold. Additionally, [20] presents a QoS-forecasting-based flow-control scheme for multi-connectivity scenario to reduce SCGFailure probability. Other solutions aim to mitigate the impact of SCGFailures: [21] proposes a fast data recovery algorithm to minimize the data interruption period caused by SCGFailures, while [22] employs packet duplication to combat

interruptions. [23] and [24] propose a cell blocking algorithm to optimize UE power consumption during frequent SCGFailures. Different from these solutions, SFM proposed in this work aims to detect and resolve problematic SCGFailure handling. Therefore, the aforementioned solutions are complementary to SFM and can be used together to improve the overall user experience on failure handlng. There are other works related to SCGFailure. [25] uses SCGFailure information as an indicator of no 5G coverage and stops the 5G scanning to save power. [26] extends the concept of SCGFailure to the beam level and proposes secondary cell beam failure recovery, which reports the failed beam to PCell and conducts beam recovery. These latter works are not directly related to handling failed SCG, and thus fall outside the scope of this paper.

VI. Conclusion

In this work, we conducted an in-depth measurement study to characterize how 5G networks handle secondary radio access failures in the US. Although such failures are not common, most failure instances are not handled properly in three forms (**U**, **M**, **R**), resulting in unnecessary and significant performance degradation. We identified root causes of each type of problematic SCGFailure handling, and proposed a device-side solution SFM to improve SCGFailure handling. Our trace-driven evaluation demonstrates that SFM can help UE avoid more than half of problematic SCGFailure instances.

There are still some remaining issues, including but not limited to measuring and understanding performance impacts on popular streaming and latency-sensitive applications, and designing cross-layer or higher-layer algorithms (on TCP congestion control and application) to further mitigate the negative impacts of SCGFailures. Last but not least, we would like to highlight that problematic SCGFailure handling significantly hurts performance because 5G currently uses non-standalone (NSA) with 5G as secondary radio access. Performance impacts of problematic SCGFailure handling should be much smaller when 5G advances to standalone (SA) and serves as master radio access. However, problematic failure handling may occur with master radio access which will not only hurt data performance but also access availability (access is interrupted with such failures).

References

[1] 3GPP, "Carrier aggregation on mobile networks," 2022. [Online]. Available: https://www.3gpp.org/technologies/carrier-aggregation-on-mobile-networks

[2] 3GPP, "TS36.101: E–UTRA; user equipment (UE) radio transmission and reception," Apr. 2023, v16.16.0.

[3] 3GPP, "TS38.331: NR; radio resource control," Mar. 2023, v16.12.0.

[4] 3GPP, "TS37.340: NR; multi-connectivity; overall description; stage- 2," Jan. 2023, v16.12.0.

[5] OpenSignal, "5G user experience report USA," 2023. [Online]. Available: https://www.opensignal.com/reports/2023/07/usa/mobile-network-experience-5g

[6] Ericsson, "How to tackle fast recovery from radio link failure," 2020. [Online]. Available: https://www.ericsson.com/en/blog/2020/9/fast-recovery-from-radio-link-failure

[7] "5G SCGFailure datasets," 2024. [Online]. Available: https://github.com/mssn/scgfailure

[8] Y. Liu and C. Peng, "A close look at 5G in the wild: Unrealized potentials and implications," in *Proc. 2023 IEEE Conf. Comput. Commun.*, 2023, pp. 1–10.

[9] MobileInsight, 2022. [Online]. Available: http://www.mobileinsight.net

[10] Y. Liu, J. Huang, and C. Peng, "The sky is not the limit: Unveiling operational 5G potentials in the sky," in *Proc. 2024 IEEE/ACM 32nd Int. Symp. Qual. Serv.*, 2024, pp. 1–10.

[11] Y. Liu, G. Guo, and C. Peng, "Demystifying secondary radio access failures in 5G," in *Proc. 25th Int. Workshop Mobile Comput. Syst. Appl.*, 2024, pp. 114–120.

[12] Z. Zhang, Y. Liu, Q. Li, Z. Liu, C. Peng, and S. Lu, "Dependent misconfigurations in 5G/4.5G radio resource control," *Proc. ACM Netw.*, vol. 1, pp. 1–20, 2023.

[13] A. Tarrias, S. Fortes, and R. Barco, "Failure management in 5G RAN: Challenges and open research lines," *IEEE Netw.*, vol. 37, no. 5, pp. 215–222, Sep. 2023.

[14] Y. Li, Q. Li, Z. Zhang, G. Baig, L. Qiu, and S. Lu, "Beyond 5G: Reliable extreme mobility management," in *Proc. Annu. Conf. ACM Special Int. Group Data Commun. Appl., technol., Architectures, Protoc. Comput. Commun.*, 2020, pp. 344–358.

[15] Z. Zhang et al., "Movement-based reliable mobility management for beyond 5G cellular networks," *IEEE/ACM Trans. Netw.*, vol. 31, no. 1, pp. 192–207, Feb. 2023.

[16] Y. Li et al., "A nationwide study on cellular reliability: Measurement, analysis, and enhancements," in *Proc. Annu. Conf. ACM Special Int. Group Data Commun. Appl., technol., Architectures, Protoc. Comput. Commun.*, 2021, pp. 597–609.

[17] K. Boutiba, M. Bagaa, and A. Ksentini, "Radio link failure prediction in 5G networks," in *Proc. 2021 IEEE Glob. Commun. Conf.*, IEEE, 2021, pp. 1–6.

[18] M. A. Islam, H. Siddique, W. Zhang, and I. Haque, "A deep neural network-based communication failure prediction scheme in 5G RAN," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 2, pp. 1140–1152, Jun. 2023.

[19] A. B. Belguidoum, M. L. Tounsi, and S. Mekaoui, "Optimization of 5G retainability and mobility in non standalone and standalone mode," in *Proc. 2nd Int. Conf. Adv. Elect. Eng.*, 2022, pp. 1–6.

[20] X. Ba, "QoS-forecasting-based intelligent flow-control scheme for multi-connectivity in 5G heterogeneous networks," *IEEE Access*, vol. 9, pp. 104304–104315, 2021.

[21] C. Pupiales, D. Laselva, and I. Demirkol, "Fast data recovery for improved mobility support in multiradio dual connectivity," *IEEE Access*, vol. 10, pp. 93674–93691, 2022.

[22] J. Rao and S. Vrzic, "Packet duplication for URLLC in 5G: Architectural enhancements and performance analysis," *IEEE Netw.*, vol. 32, no. 2, pp. 32–40, Mar./Apr. 2018.

[23] K. K. Jha, N. A. K. Jangid, R. P. Kamaladinni, N. P. Shah, and D. Das, "Efficient algorithm to reduce power consumption for EUTRA-new radio dual connectivity RAN parameter measurements in 5G," in *Proc. IEEE 3rd 5G World Forum*, 2020, pp. 536–541.

[24] T. Ou, Q. Zhang, and X. Sun, "Dynamic blocking of 5G cells in non-standalone networks," *Tech. Discl. Commons*, 2021.

[25] A. K. Jangid, N. K. K. Jha, and D. Das, "Efficient protocol for EUTRA new radio dual connectivity handling based on location," in *Proc. 2020 IEEE 3rd 5G World Forum*, IEEE, 2020, pp. 103–108.

[26] D. Kurita and T. M. K. Harada, "5G advanced technologies for mobile broadband," *NTT DOCOMO Tech. J.*, vol. 22, no. 3, pp. 90–105, 2021.

Yanbing Liu (Graduate Student Member, IEEE) received the MS and BS degree in Department of Electronic Engineering and Information Science from the University of Science and Technology of China. He is currently working toward the PhD degree in Department of Computer Science with Purdue University. His research interests are in the area of mobile networking, with a focus on 5G/6G networks measurement and design.

Chunyi Peng (Senior Member, IEEE) received the PhD in Department of Computer Science, from the University of California, Los Angeles, in 2013. She is currently an associate professor with the Department of Computer Science, Purdue University, West Lafayette, IN, USA. Prior to that, she worked as an assistant professor with the Department of Computer Science and Engineering, Ohio State University and an associate researcher at Microsoft Research Asia. Her research interests are in the broad areas of mobile networking, system and security, with a recent focus on renovating 5G access technologies, AI for networks, mobile network security, mobile edge computing mainly for autonomous drones and robots.