Reminis: A Simple and Efficient Congestion Control Scheme for 5G Networks and Beyond

Parsa Pazhooheshy
University of Toronto
parsap@cs.toronto.edu

Soheil Abbasloo
University of Toronto
abbasloo@cs.toronto.edu

Yashar Ganjali
University of Toronto
yganjali@cs.toronto.edu

***Abstract*—5G cellular networks are expected to support diverse emerging services with stringent delay requirements, ranging from 20 ms for virtual reality application to 100 ms for immersive video streaming. However, the highly variable and unpredictable nature of 5G access links poses challenges to existing end-to-end (e2e) congestion control (CC) schemes, resulting in suboptimal performance. This paper demonstrates that a simple yet effective e2e Transport Layer CC scheme for 5G networks can be achieved by blending non-deterministic exploration techniques with straightforward proactive and reactive measures. Our proposed scheme, Reminis, is designed to achieve high controllable performance while possessing provable properties. Through extensive experiments on emulated and real-world 5G networks from different vendors, with different network configurations and mobility scenarios, we illustrate the performance benefits of Reminis in 5G networks compared to state-of-the-art CC schemes and its generalizability to 3G and 4G networks. For example, in 5G standalone scenarios, Reminis achieves a 2.2 times lower 95th percentile delay compared to a recent design by Google, BBR2, while maintaining the same link utilization.**

***Index Terms*—5G Network, Congestion Control, Cellular Networks, Transport Layer**

I. Introduction

Congestion Control (CC), as one of the active research topics in the network community, has played a vital role during the last four decades in satisfying the quality of service (QoS) requirements of different applications [1]–[9]. Although most of the early efforts for designing CC schemes targeted general networks with their general characteristics, as time passed and new network environments emerged, designing environment-aware CC schemes showed its advantages.

One of the important emerging network environments with huge potential is the 5G cellular network. In the first quarter of 2022, the number of connections over 5G reached more than 700 million, while it is expected that by the end of 2026, this number will exceed 4.8 billion worldwide. Considering such a huge increase in the number of 5G users, the variety of current and future applications, and the range of new network characteristics and challenges it brings, the need for a 5G-tailored CC scheme reveals itself.

ISBN 978-3-903176-72-0 ©2025 IFIP

A. What Makes 5G Different?

Designing CC schemes for 5G networks presents several challenges not encountered in previous generations such as 4G. However, among these challenges lie opportunities that a CC scheme tailored for 5G can leverage. Here, we outline some of these challenges and opportunities, and later in Sec. I-C, we delve into how they influence CC design decisions.

Challenges

C1 – Significantly Larger Bandwidth-Delay Product: 5G networks, especially those operating in millimeter wave bands, offer impressive link capacities of approximately 1 to 2 Gbps with end-to-end delays around 20 ms [10], [11]. In comparison, data center networks with 100 Gbps links and 10 µs delays have a much lower bandwidth-delay product (BDP), making 5G's BDP roughly 40 times larger on average. Even against 4G networks, typically 20M bps links and 40 ms delays, the BDP of 5G is about 50 times higher.

C2 – Highly Variable Unpredictable Access Links: A defining feature of 5G is its wide-ranging capacity fluctuations. Although capable of reaching up to 2 Gbps, 5G links can quickly drop to 4G speeds or close to zero in "5G dead zones" [11]. For example, previous studies report a standard deviation of approximately 432 Mbps in 5G link capacities, driven by factors such as user mobility, obstacles, network coverage, cell size, and handover processes.

C3 – Emerging Applications with Unique Delay Requirements: 5G is enabling a host of delay-sensitive applications, including AR/VR, online gaming, vehicle-to-vehicle communication, tactile Internet, remote medical operations, and machine learning services. These applications have diverse delay requirements, from as low as 20 ms for AR/VR to 40–60 ms for cloud gaming and up to 100 ms for immersive video streaming.

Opportunities

O1 – More Stable RAN Latency: An encouraging aspect of 5G is the relatively stable latency within its Radio Access Network (RAN) compared to LTE [12]. This stability reduces noise in end-to-end RTT signals, making it easier to leverage simple delay-based metrics

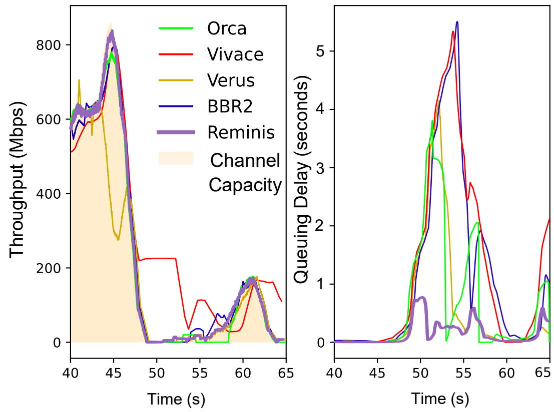

Fig. 1. Performance of state-of-the-art CCA on a slice of a sample 5G trace gathered by prior work [11].

to infer network congestion, a strategy effectively employed by Reminis.

O2 – Mobile Edge Computing: The integration of 5G with edge-based frameworks, such as Mobile Edge Computing (MEC), along with 5G network slicing, presents new design opportunities. In these edge-based architectures, where users operate on isolated logical networks atop shared infrastructure, traditional concerns about TCP-friendliness are reduced, paving the way for CC schemes specifically tailored for 5G.

B. *Impact of 5G's Unique Properties on CC*

As a motivating experiment, we use a 5G cellular trace from previous work [11] to evaluate the delay and throughput of state-of-the-art CC schemes—BBR2 [2] (white-box), Verus [7] (4G-tailored), Orca [13] (RL-based) and Vivace [8] (online learning) - in 20 seconds (see Sect. V-A). The experiment aims to illustrate how the challenges discussed in Sec. I-A impact these schemes.

Fig. 1 shows that the capacity of the 5G link drops from nearly 1 Gbps to zero in just 3 seconds, exemplifying the high variability of the 5G links. In this scenario, BBR2 suffers from severe delay - accumulating more than 5 seconds of queueing delay - because its model-based assumptions do not hold under such unpredictable conditions. Verus, designed for 4G networks, is too slow to adjust to rapid changes, resulting in low utilization and a long queueing delay. Similarly, although Orca outperforms both BBR2 and Verus, it still encounters significant queueing delays, likely due to generalization issues inherent in RL-based designs. Vivace, despite employing online learning to avoid extensive offline training, cannot keep pace with the fast-varying network conditions; for instance, after a capacity drop of around 47 seconds, it takes over 10 seconds to reduce its sending rate below the new link capacity. This experiment demonstrates that current CC schemes struggle with the rapid and unpredictable dynamics of 5G networks, underscoring the need for a tailored solution like Reminis.

C. *Design Decisions*

5G networks expose the shortcomings of existing congestion control (CC) methods, such as BBR2, Verus, and Orca, especially for delay-sensitive applications. To address this, we present Reminis, a high-performance CC scheme tailored for 5G. It rapidly capitalizes on sudden bandwidth spikes, mitigates delay surges caused by dead zones, and remains simple to deploy. Instead of complex learning-based techniques, Reminis relies on intuitive heuristic-driven control rules within a standard delay-target framework.

A core innovation is *nondeterministic exploration*, which dynamically adjusts the congestion window (CWND) when the delay is below the target, supported by both mathematical proofs and experimental results. Reminis also features *fast proactive* and *agile reactive* slowdowns that use simple delay statistics to predict and alleviate congestion before it harms throughput. Additional modules handle dead zones and other erratic network conditions. Finally, by combining lightweight acknowledgment-based AIMD logic with a periodic delay-based loop, Reminis maintains high utilization with minimal CPU overhead, ideal for resource-constrained 5G devices, while delivering high throughput and very low end-to-end delay.

D. *Contributions*

The main contributions of this paper are as follows. First, we show that strong performance on 5G links does not require complex learning techniques; instead, Reminis leverages tailored methods such as non-deterministic exploration and proactive slowdown to address 5G-specific challenges. Second, we provide a mathematical proof of Reminis' stability, demonstrating its convergence to a steady state with a bounded, controllable self-inflicted queuing delay. Third, our evaluation spans both emulated 5G traces from various providers and real-world tests on a North American 5G network under various mobility conditions (stationary, walking, and driving), proving Reminis' adaptability to unpredictable 5G links. Fourth, by extending our evaluation to 3G and 4G networks, we find that while Reminis continues to outperform other schemes, the performance gap is narrower, highlighting the unique challenges of 5G. Finally, our work led to debugging and improving Mahimahi, previously unable to accurately emulate high-capacity 5G links, with our publicly available patch supporting further research on 5G congestion control.

The rest of this paper is organized as follows. Sec. III outlines Reminis's high-level design for 5G networks. Sec. IV describes each component in detail. Sect. V evaluates performance in both emulated and real-world environments, including legacy 3G/4G. Finally, Sec. VI

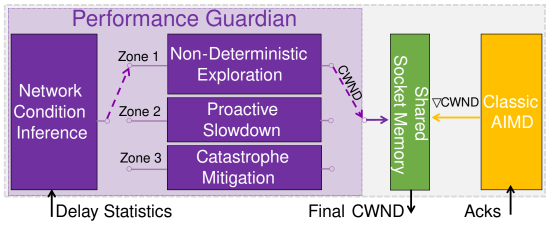

Fig. 2. Reminis High-Level Block Diagram.

examines Reminis's behavior under varying network parameters and its CPU utilization.

II. Related Works

General Purpose CC Schemes: Since the early days of congestion control designs, various schemes such as TCP Reno, TCP NewReno, TCP CUBIC and BIC have been proposed [1], [14]–[16]. These legacy loss-based CC schemes, along with delay-based variants like TCP Vegas [3], operate under the assumption of fixed-capacity links, which does not fit the dynamic nature of cellular networks. Recent efforts like Copa [5] and LEDBAT [17] aim to address this disparity.

Environment-Aware CC schemes: Several works focus specifically on cellular networks. For example, Sprout [4] uses packet arrival times combined with probabilistic inference, but its efficiency in 5G networks is questionable. Similarly, Verus [7] attempts to create a delay profile of the network, but lacks agility to adapt to 5G networks. Other recent cellular-tailored works such as C2TCP [18] and ExLL [19] show potential but struggle to adapt to rapid surges in 5G links. Works like ABC [20] utilizes the feedback from the routers to adjust the sender's CWND value in a wireless setting. However, deploying such in-network feedback approaches requires new devices or updates to existing ones, posing challenges, especially in cellular networks. Another set of cellular-tailored CC schemes relies on physical layer measurements, exemplified by PBE-CC [21], and CQIC [22]. However, these approaches raise concerns regarding power consumption and privacy due to their need for access to end hosts' physical layer information.

ML-based CC Schemes: ML-based approaches, such as Remy [23] and PCC-Vivace [8], leverage offline or online learning to optimize sending rates. However, these methods face challenges in accurately modeling the highly variable nature of 5G networks. ML-based schemes such as DeepCC [6] target cellular networks but struggle to control end-to-end delay effectively in 5G settings. Furthermore, interpretability issues and generalization concerns hinder the applicability of ML-based schemes in new environments.

III. Design Overview

As shown in Fig. 2, Reminis consists of two main components. The first is the AIMD unit, an ACK-triggered mechanism that performs the standard AIMD behavior upon receiving acknowledgments. The second is the Performance Guardian (or Guardian) module, which runs periodically to adjust the congestion window (CWND) to maintain a low delay while achieving high throughput.

At the start of each period, the Guardian employs a two-step process. First, its Network Condition Inference (NCI) module uses simple end-to-end RTT samples - analyzing both the delay and its first-order derivative - to infer the current network condition. Due to stable RAN latency, these RTT signals are reliable in distinguishing different conditions. Based on the inferred state, the Guardian then selects one of three modules: Non-Deterministic Exploration (NDE), Proactive Slowdown (PS), or Catastrophe Mitigation (CM). If there is potential to gain more throughput, the Guardian activates NDE to quickly explore additional available bandwidth. If an unwanted delay surge is anticipated, PS is triggered to proactively reduce CWND and mitigate future delay increases. Should proactive measures fail and delays already spike significantly, CM is employed as a reactive measure to sharply reduce CWND and prevent further delay escalation. Sec. IV provides further details on these modules.

Why Guarding Periodically and not Per Packet? There are two key reasons for the Guardian's periodic operation. First, due to the high variability of 5G networks (as noted in C2, Sec. I-A), relying on per-packet statistics can be misleading because single-packet measurements are prone to noise. By sampling multiple packets over defined intervals, called sampling intervals (SI), Reminis obtains a more stable and accurate view of network conditions. Second, given the high capacities of 5G access links, making compute-heavy decisions on a per-packet basis would significantly increase CPU utilization. Periodic logic, by contrast, reduces this computational burden while still capturing the necessary network dynamics.

The Role of AIMD Block: While periodic guarding offers stability by aggregating multiple samples, relying solely on it can reduce system agility in highly variable networks. Purely periodic logic may react too slowly to sudden changes in link capacity during each sampling interval. To address this, Reminis incorporates a classic AIMD block that provides per-packet reactions within each SI. This hybrid approach improves agility (see Fig.12) and overall performance (Fig.3), while maintaining very low CPU overhead, which is essential for congestion control in high-capacity 5G networks.

IV. Reminis Design

In this section, we discuss the main components of the Guardian block. First, we introduce the NCI module responsible for inferring network conditions based on delay statistics. As discussed in O1, delay measurements are less noisy in 5G networks, hence they are sufficient for inferring sufficient network conditions for a CC scheme. Then, we describe the modules responsible for modifying the CWND namely NDE, PS, and CM to mitigate C1, C2, and C3. The way these modules are put together makes Reminis capable of contending with high volatility in 5G access link capacity and keeping the e2e delay low.

A. Network Condition Inference (NCI)

The NCI module infers network conditions by leveraging two signals: the end-to-end RTT delay and its derivative. Delay is measured directly from RTT samples, and because the Guardian runs once every sampling interval (SI), the delay derivative is defined as the difference between two consecutive delay measurements divided by the time difference between SIs. Each SI is set to the observed minimum RTT ($mRTT$)—the lowest RTT seen since the flow began, which may not match the actual network minimum.

Reminis aims to keep the delay below a target called the delay tolerance threshold (DTT). This threshold can be defined by the target application or set to a default of $1.5 \times mRTT$. Using the DTT as a reference, the NCI module analyzes the delay signals to determine the current congestion state during each SI. These inferred conditions then guide the activation of one of the following modules: Non-Deterministic Exploration (NDE), Proactive Slowdown (PS), or Catastrophe Mitigation (CM).

To maintain simplicity and a lightweight design, the NCI module classifies network conditions into three congestion zones based on the delay (d_n) and its derivative (∇d_n) measured in the current SI (SI_n). Zone 1 is defined as $d_n \leq DTT$ and $\nabla d_n \leq 0$. In this zone, the delay is below DTT and is decreasing, indicating that the sending rate is less than the channel capacity and the queue is draining; this creates an opportunity to explore higher throughput. Zone 2 is defined as $d_n \leq DTT$ and $\nabla d_n > 0$. Although the delay is still below the DTT, its increase suggests that the queue is building up, so maintaining or increasing the current congestion window (CWND) could risk violating the delay target. Zone 3 is defined as $d_n > DTT$, indicating that the delay has exceeded the threshold; in this case, the sender should dramatically reduce its CWND. Transitions into Zone 3 can occur for various reasons, including entering 5G dead zones.

To quantify how much the delay exceeds DTT, the Guardian employs a function called `SafeZone`, defined as follows:

$$\texttt{SafeZone}(d) = 1 - \frac{d - mRTT}{DTT - mRTT}. \qquad (1)$$

Based on the zone determined by the NCI module, one of the NDE, PS, or CM modules is activated.

B. Non-Deterministic Exploration (NDE)

As discussed earlier, being in Zone 1 indicates that there is room for sending more packets to achieve higher throughput. However, factors such as user mobility, dynamic physical obstacles, and the wireless scheduler's resource allocation make it challenging to accurately determine the extent of a safe increase. In response, the NDE module within Reminis is tasked with discovering and utilizing the available capacity without risking bufferbloat. When Zone 1 is identified by the NCI module, the Guardian triggers the NDE module to explore different Congestion Window (CWND) values in a non-deterministic manner. This approach effectively addresses the unpredictability of available link capacities, enabling Reminis to take advantage of sudden surges in access link capacity.

However, it is crucial to avoid uncontrolled queue buildup during exploration. To that end, the NDE module controls the average of the stochastic decision-making process based on the observed trend in link capacity. In particular, exploration becomes more aggressive when the Guardian measures high negative delay derivatives, since a more negative derivative indicates that the sending rate is far below the channel capacity.

To implement this, the NDE module maintains a Gaussian distribution, $\mathcal{N}(\mu_n, \sigma_n^2)$, whose parameters are updated at every sampling interval (SI) according to the rules: $\mu_n \leftarrow \mu_{n-1} - \nabla d_n$ and $\sigma_n^2 = \frac{\mu_n}{4}$. Upon activation, the NDE module draws a sample $x \sim \mathcal{N}(\mu_n, \sigma_n^2)$ and feeds it into a Sigmoid function defined by $S(x) = \frac{1}{1+e^{-x}}$.

The output of this Sigmoid function is then used to increase the current CWND by multiplying it by $2^{S(x)}$, as shown in the following equation: $cwnd_n \leftarrow cwnd_n \times 2^{S(x)}$. Since the Sigmoid function ranges between 0 and 1, the NDE module increases the CWND by a factor between 1 and 2. When Reminis consistently measures negative delay derivatives, the incremental factor approaches 2, enabling rapid adaptation to an increase in link capacity. Conversely, if a negative delay derivative occurs after several positive ones, the exploration is more conservative, yielding only a slight increase above 1.

We prove that the NDE module accelerates the convergence of the AIMD logic. Specifically, assuming that w_1 is the CWND that fully utilizes a fixed link without causing queue build-up, we show that: Reminis helps AIMD logic reach w_1 in $\mathcal{O}(\log w_1)$ instead of $\mathcal{O}(w_1)$ during the congestion avoidance phase. Due to space limitations, the proof is omitted.

C. Proactive Slowdown (PS) and Catastrophe Mitigation (CM)

Considering the high fluctuations of 5G access links, the primary role of the PS and CM modules is to effectively control the end-to-end delay without causing significant underutilization.

Proactive Slowdown (PS): This module is activated whenever the NCI infers Zone 2. In Zone 2, Reminis must act prudently to prevent any violation of the delay tolerance threshold (DTT) in the next SI. The PS module decreases the current CWND if the delay approaches DTT. To predict the risk of a DTT violation, PS calculates the expected delay in the next SI using a first-order regression predictor: $d_{n+1} = d_n + \nabla d_n \times SI$.

The following equation shows how the PS module decreases the CWND upon activation. The module reduces the CWND if the predicted delay in the next SI exceeds DTT, and it does so more aggressively as the expectation for DTT violation increases: $cwnd_n \leftarrow cwnd_n \times 2^{\min(0,\, \mathtt{SafeZone}(d_{n+1}))}$.

Catastrophe Mitigation (CM): Despite the actions of the PS module, sudden decreases in link bandwidth can still force Reminis into Zone 3. In such cases, it is imperative to reduce the delay as quickly as possible to meet the application's requirements. Therefore, when the NCI infers Zone 3, the CM module is activated. CM reduces the CWND by at least half, the reduction being more aggressive on the basis of how much the current delay exceeds DTT. The CWND update rule for the CM module is given as follows: $cwnd_n \leftarrow cwnd_n \times 2^{\mathtt{SafeZone}(d_n)} \times 0.5$.

D. Reminis Steady State

One of our main goals was to design an interpretable CC scheme with provable properties. Considering that, we mathematically prove that

Theorem 1: On average, Reminis converges to a steady state with a queueing delay of no more than $(1 + S(\frac{\ln 4 - 1}{2BDP}) \ln 2) q_{th}$.

In the above Theorem, S is the Sigmoid function as introduced in Sec. IV-B, BDP is the bandwidth-delay product of the network, and q_{th} is the DTT. Due to space limitations, the detailed proof of the above Theorem is omitted and could be added to the final version of the paper.

V. General Evaluations

In this section, we thoroughly evaluate Reminis and compare it with other state-of-the-art e2e CC schemes in trace-based emulations and field experiments. The emulations help us measure Reminis performance over various scenarios, whereas the in-field tests help us verify Reminis performance in a much more complex real-world network.

Metrics: The main metrics used in this paper, suitable for a real-time application, are the average throughput (or equivalently link utilization) and delay-based statistics such as average and 95th percentile packet delays.

Compared CC Schemes: We compare Reminis with different classes of state-of-the-art e2e CC schemes. The first class is general purpose CC schemes such as TCP CUBIC [1], Google's BBR2 [2], TCP Vegas [3], and Copa [5]. The second class is CC algorithms that are custom-designed for cellular networks. These schemes are C2TCP [18], Verus [7], and Sprout [4]. The final class is learning-based CC schemes. We compare Reminis with DeepCC [6], targeting cellular networks, and PCC-Vivace [8] as a general purpose learning-based CC scheme.

A. Trace-based Emulations

Mahimahi's Limitation in High-Speed Scenarios: Mahimahi [26] was not originally designed for high-throughput emulations (e.g. 5G). During Reminis evaluations (Sec. V-A), we pinpointed its single-threaded logging as the primary bottleneck. By revising Mahimahi to dedicate individual threads for logging and adjusting TUN/TAP settings, we significantly improved its performance.

Setup: We use trace-driven emulations to evaluate Reminis and compare it with other CC schemes under reproducible network conditions. We use our patched Mahimahi as the emulator and the 5G traces collected by prior work [11] as our base network traces. After patching Mahimahi, we evaluated the general performance of Reminis and other relevant CC schemes. For these experiments, we used 60 different 5G traces gathered in North America by previous work [11] in various scenarios. Each run is set to be 3 minutes, and we repeat each run 3 times. For these experiments, we fix the minimum intrinsic RTT of the network to 20 ms based on prior measurements performed by [10]. Furthermore, since currently there are two different deployments of 5G networks, we consider two different settings for bottleneck bandwidth size. The first deployment is the Non-Standalone (NSA) mode, where operators are reusing the legacy 4G infrastructure to reduce costs. In NSA mode, we expect to have 4G-tuned buffers, which would be smaller than the 5G-tuned buffers. For the NSA mode, based on the measurements made in previous work [10], we set the buffer size to 800 packets. The second deployment is the standalone (SA) version, where the network infrastructure is also changed to recognize the needs of 5G networks. In this case, we configure the buffer size to 3200 packets [10].

Standalone (SA) Scenario: Fig. 3 (left column) presents the performance of various congestion control schemes in the SA experiment. The dashed curve shows the performance of Reminis with different DTT values, while the larger star marks its default setting ($DTT = 1.5 \times mRTT$). (Sect. VI-B further examines the sensitivity of Reminis to this parameter.)

Pure loss-based schemes such as CUBIC fully utilize the link by occupying the bottleneck buffer but incur

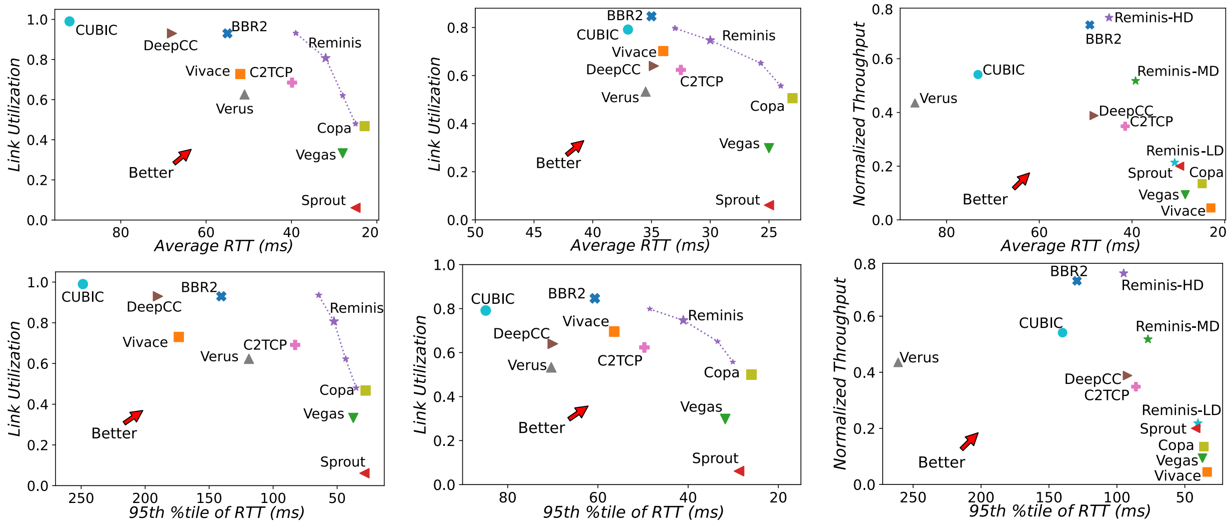

Fig. 3. Throughput-Delay for SA (left column), NSA (middle column) and In-Field (right column) Emulations.

high end-to-end delays. In contrast, Reminis strikes a balance between delay and throughput. On average, default Reminis achieves a 5× reduction in the 95th percentile delay compared to CUBIC, with only a 20% decrease in link utilization. When the DTT value increases to $2 \times mRTT$, making Reminis more throughput-focused, it attains a 2.2× lower 95th percentile delay than BBR2 while maintaining the same link utilization.

Furthermore, delay-based algorithms like Vegas and Copa exhibit poor link utilization. Across all experiments, default Reminis provides about 2.42× more throughput than Vegas, with its 95th percentile and average delays being only 1.4× and 1.14× higher, respectively.

The key takeaway is that in SA scenarios with deep buffers, throughput-hungry schemes such as CUBIC or BBR2 can fully utilize the link at the expense of significant delays. However, Reminis uses its proactive slowdown and catastrophe mitigation modules to effectively control delay. Without its Non-Deterministic Exploration module, Reminis would struggle like other delay-based schemes. Together, these modules enable Reminis to achieve an optimal delay-throughput trade-off.

Non-Standalone (NSA) Scenario: The middle column of Fig. 3 shows the performance of all CC schemes in the NSA scenario. Due to the small buffer size, even throughput-hungry schemes such as BBR2 and CUBIC achieve only about 80% link utilization. In this setting, Reminis strikes an ideal balance in the delay-throughput trade-off: while matching CUBIC's utilization, default Reminis delivers a 2× lower 95th percentile delay. Overall, the performance trends mirror those in the SA scenario, with generally reduced delays and utilizations across all schemes because of the limited buffer size.

B. In-Field Evaluations

Real-world cellular networks are more complex than the emulated ones due to factors such as other users and the varying behavior of the packet schedulers of the base station. We evaluated Reminis on 5G networks deployed in North America using servers as senders, a 5G SIM card, and a Samsung Galaxy S20 5G as client. In our experiments, the mRTT ranged from 20 to 30 ms, and we conducted 80 experiments per congestion control scheme, with each run lasting 15 seconds. These tests were performed at various times and locations under both stationary and walking conditions to capture diverse network dynamics.

In stationary scenarios, access link capacity fluctuates primarily due to changes in line of sight (LoS) and variations in the wireless scheduler. Even minor obstructions, such as a human body, can trigger 5G-to-4G handoffs that lead to performance degradation. In addition, differences in historical resource usage and the number of active users can result in varying available capacities. The resource demands in different 5G slices can also change unpredictably, suddenly altering the network resources available to a user.

For each experiment, the throughput of the schemes is normalized to the maximum achieved in that specific scenario. We tested three versions of Reminis: Reminis-HD with a DTT of 60 ms, Reminis-MD with a DTT of 40 ms and Reminis-LD with a DTT of 30 ms. The right column of Fig. 3 summarizes the in-field results: Reminis-HD matches the throughput of BBR2 while reducing the 95th percentile delay by an average of 1.47×, and it outperforms TCP CUBIC by achieving 1.34× higher throughput and 1.4× lower 95th percentile RTT. With a tighter DTT, Reminis-MD achieves the same

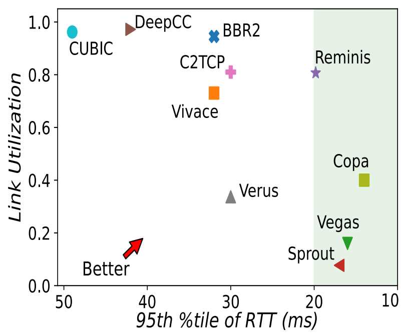

Fig. 4. MEC-Flavoured Exp.

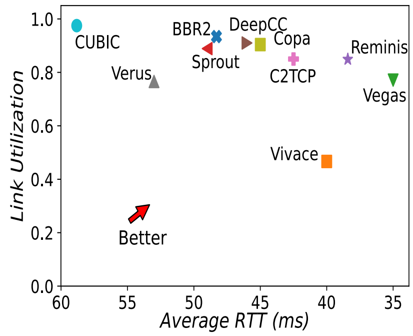

Fig. 5. Experiments on 4G Traces.

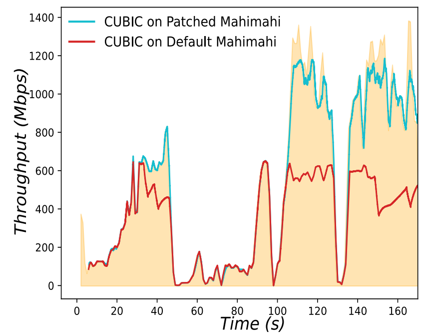

Fig. 6. Mahimahi Patch.

throughput as CUBIC while lowering the 95th percentile delay by 1.7×. In general, in-field evaluations closely mirror the NSA emulation results, which confirms our assumptions.

C. MEC-Flavored Emulations

We emulate a 5G MEC scenario by setting the intrinsic RTT to 10 ms for a VR application with a 20 ms delay requirement, while keeping the rest of the setup representative of SA conditions. As shown in Fig. 4, only four CC schemes achieve the AR/VR latency goal, but Reminis delivers at least 2.4× more throughput than the others, thanks to its Non-Deterministic Exploration module. Meanwhile, Proactive Slowdown and Catastrophe Mitigation help Reminis stay within the 20 ms delay target, illustrating the effectiveness of its design choices.

D. Reminis in 3G and 4G Networks

Here, we show that mechanisms utilized by Reminis are effective to maintain the e2e delay low and adapt to the channel capacity variations not only in 5G cellular networks but also in other networks such as 3G and 4G cellular networks.

To do this, we use various 3G and 4G traces (gathered, respectively, by [18] and [6]) and evaluate the performance of different schemes. First, Reminis still performs very well in both 3G and 4G scenarios. For instance, compared to BBR2, on 4G and 3G traces, Reminis achieves a 1.48× and 1.33 × lower average queuing delay, respectively, while BBR2's throughput is only 1.1 × and 1.05 × more than Reminis. Second, the performance gap between other CC schemes and Reminis in 3G and 4G scenarios is smaller compared to 5G scenarios. The main reason for that is the fact that 5G networks have an order of magnitude larger BDP, deeper buffers, and more volatile access links compared to 3G and 4G networks. This means in the 5G setting, wrong actions of a CC scheme have higher chances of being penalized more and manifest in performance issues.

VI. Deep Dive Evaluations

In this section, we will look under the hood and investigate the dynamics of Reminis and the role of its components. We will also investigate the impact of different parameters such as intrinsic RTT, buffer size, and DTT on Reminis. Finally, we will end this section by examining Reminis' fairness and overhead aspects.

A. Impact of Buffer Size

In cellular networks, each user has its own base station buffer, which reduces packet drops, but can cause buffer overload [27] and self-inflicted delays [4]. We vary the buffer size in our emulated network from 800 to 51200 packets; the lower bound (800 packets) is based on the measurements of the NSA-5G network [10]. Fig. 8 shows that CUBIC tends to fill available buffers, resulting in higher delays with larger buffer sizes. In contrast, Reminis maintains delay near its Delay Tolerance Threshold (DTT) of 30 ms by controlling its congestion window appropriately. Moreover, Fig. 7 indicates that Reminis achieves approximately 80% link utilization across all buffer sizes. This consistent performance is critical, especially since NSA-5G networks often have smaller buffers similar to those of 4G systems.

B. Impact of Delay Tolerance Threshold

Delay Tolerance Threshold (DTT) is a key parameter in Reminis' design and performance. Reminis will become more conservative when the measured delay exceeds DTT, so we expect to have a trade-off between delay and link utilization based on different values of DTT. Large DTT values steer Reminis toward being more throughput-oriented, while small DTT values guide Reminis toward being more delay-oriented. In Fig. 9, Reminis-X means a version of Reminis where the DTT parameter has been set to X ms.

To provide context, we include TCP Vegas, known for prioritizing delay, and TCP CUBIC, renowned for maximizing throughput, in Fig. 9. Furthermore, CUBIC is accompanied by CoDel and Pie, two AQM schemes, although unable to utilize the link beyond 60%. This limitation underscores the challenge of AQM schemes on 5G links.

Observing Fig. 9, we notice the expected impact of DTT on Reminis' performance. A larger DTT encour-

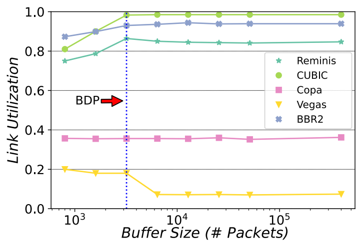

Fig. 7. Buffer Size vs. Throughput.

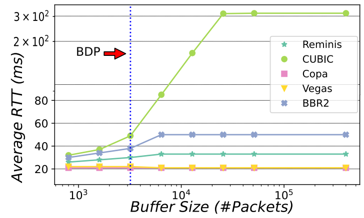

Fig. 8. Buffer Size vs. Delay

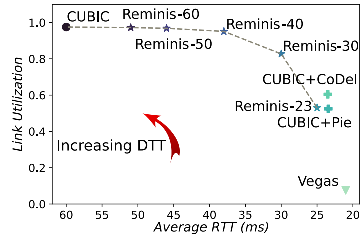

Fig. 9. Impact of DTT.

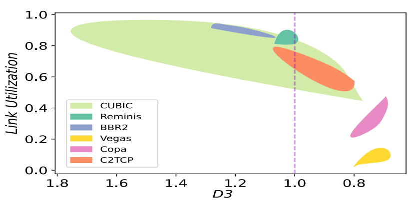

Fig. 10. Impact of min. delay.

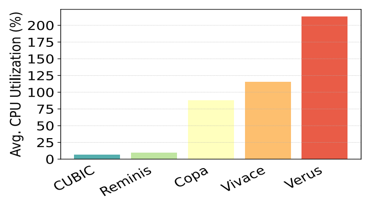

Fig. 11. Reminis' CPU Utilization.

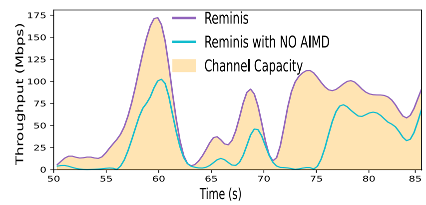

Fig. 12. Impact of AIMD Block.

ages Reminis to prioritize throughput, while a smaller DTT emphasizes delay sensitivity. In particular, Reminis achieves its objectives without sacrificing significant throughput. For example, with DTT = 300 ms, Reminis achieves its target with only a 20% reduction in link utilization compared to CUBIC, which has an average RTT of 60 ms.

C. Impact of Network's Intrinsic RTT

The intrinsic RTT depends on factors such as the distance between the UE and the server [28]. In this experiment, we evaluated different schemes for emulated networks with intrinsic RTTs of $\{5, 10, 20, 30, 40, 50\}$ ms and corresponding DTTs of $\{7.5, 15, 30, 45, 60, 75\}$ ms. We assume an SA version with a buffer size set to the bandwidth-delay product. (Note: Reminis automatically adjusts its SI to the observed $mRTT$.)

We define the deviation parameter $D3 = \frac{d}{DTT}$, where d is the average delay; $D3 = 1$ means the delay complies with the DTT. Fig. 10 shows that in three runs per RTT value, Reminis maintains $D3 < 1.03$ and approximately 80% link utilization. In contrast, delay-based schemes like Copa or Vegas suffer from poor utilization (for example, Copa drops to 20%), while C2TCP, BBR2, and CUBIC exhibit high variability in both $D3$ and utilization, which is undesirable for cellular networks.

D. CPU Utilization Analysis

In the context of 5G's power constraints, a lightweight CC scheme is essential to minimize computational demands. We evaluated Reminis' CPU utilization alongside other schemes by transmitting traffic over a 720 Mbps emulated link for 2 minutes, aligning with 5G link capacity measurements. Fig. 11 shows the comparison. Despite Reminis' Guardian module operating in user space, it demonstrates impressive CPU performance. Reminis and CUBIC exhibit CPU utilization of around 9.2% and 6%, respectively. This efficiency stems from Reminis' simplistic design, leveraging basic delay statistics and low-overhead AIMD actions.

E. Impact of AIMD Block

To study the impact of the AIMD module, we remove it from Reminis. This modified version undergoes all experiments outlined in Sec. V-A under the SA scenario, assessing its throughput and delay.

Results indicate that removing the AIMD block results in an average loss of 15% (up to 30%) in link utilization, without yielding noticeable improvements in delay performance. Fig. 12 provides further insight, depicting a segment of a sample 5G trace where Reminis, lacking the AIMD block, struggles to adapt to fluctuations in access link capacity.

These findings underscore the significance of the AIMD block within Reminis. The Guardian's periodic operation (one action per SI) may occasionally overlook rapid channel dynamics inherent in 5G links. Here, the AIMD block steps in, injecting fine-grained dynamics into Reminis' actions through its Ack-triggered logic, enhancing its agility. Moreover, the AIMD module increases the number of RTT samples within each SI, which contributes to more reliable average statistics. This reduction in measurement noise enables Reminis to obtain a clearer understanding of network conditions. In essence, by offering additional samples, AIMD bolsters Reminis' capability to assess network conditions more accurately.

VII. Conclusion

This study presents Reminis, an end-to-end congestion control design tailored for 5G networks. Reminis achieves high throughput and low controlled delay without the need for complex learning-based schemes or prediction algorithms. By leveraging non-deterministic throughput exploration algorithms along with proactive/reactive delay control mechanisms, Reminis effectively adapts to the highly variable nature of 5G cellular networks. Our experiments on 3G, 4G, and 5G networks, including controlled emulations and real-world scenarios, with different configurations and different mobility scenarios, demonstrate Reminis' success in meeting its design goals and outperforming state-of-the-art CC schemes on 5G networks. Importantly, Reminis boasts deployment friendliness with low overheads and does not require changes in cellular network devices.

References

[1] S. Ha, I. Rhee, and L. Xu, "Cubic: a new tcp-friendly high-speed tcp variant," *ACM SIGOPS Operating Systems Review*, vol. 42, no. 5, pp. 64–74, 2008.

[2] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "Tcp bbr v2 alpha/preview release," https://github.com/google/bbr/blob/v2alpha/README.md, 2019.

[3] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, *TCP Vegas: New techniques for congestion detection and avoidance*. ACM, 1994, vol. 24, no. 4.

[4] K. Winstein, A. Sivaraman, H. Balakrishnan *et al.*, "Stochastic forecasts achieve high throughput and low delay over cellular networks." in *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, 2013, pp. 459–471.

[5] V. Arun and H. Balakrishnan, "Copa: Practical delay-based congestion control for the internet," in *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, 2018, pp. 329–342.

[6] S. Abbasloo, C.-Y. Yen, and H. J. Chao, "Wanna make your tcp scheme great for cellular networks? let machines do it for you!" *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 265–279, 2021.

[7] Y. Zaki, T. Pötsch, J. Chen, L. Subramanian, and C. Görg, "Adaptive congestion control for unpredictable cellular networks," in *Proceedings of the 2015 ACM SIGCOMM 2015 Conference*, 2015, pp. 509–522.

[8] M. Dong, T. Meng, D. Zarchy, E. Arslan, Y. Gilad, B. Godfrey, and M. Schapira, "{PCC} vivace: Online-learning congestion control," in *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, 2018, pp. 343–356.

[9] S. Abbasloo, T. Li, Y. Xu, and H. J. Chao, "Cellular controlled delay tcp (c2tcp)," in *2018 IFIP Networking Conference (IFIP Networking) and Workshops*. IEEE, 2018, pp. 118–126. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8696844

[10] D. Xu, A. Zhou, X. Zhang, G. Wang, X. Liu, C. An, Y. Shi, L. Liu, and H. Ma, "Understanding operational 5g: A first measurement study on its coverage, performance and energy consumption," in *Proceedings of the 2020 ACM SIGCOMM 2020 Conference*, 2020, pp. 479–494.

[11] A. Narayanan, E. Ramadan, R. Mehta, X. Hu, Q. Liu, R. A. Fezeu, U. K. Dayalan, S. Verma, P. Ji, T. Li *et al.*, "Lumos5g: Mapping and predicting commercial mmwave 5g throughput," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 176–193.

[12] Y. Pan, R. Li, and C. Xu, "The first 5g-lte comparative study in extreme mobility," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 6, no. 1, pp. 1–22, 2022.

[13] S. Abbasloo, C.-Y. Yen, and H. J. Chao, "Classic meets modern: A pragmatic learning-based congestion control for the internet," in *Proceedings of the ACM SIGCOMM Conference*, 2020, pp. 632–647.

[14] D. Cox and L.-R. Dependence, "a review," *Statistics: An Appraisal, HA David and HT David (Eds.)*, pp. 55–74, 1984.

[15] T. Henderson, S. Floyd, A. Gurtov, and Y. Nishida, "The newreno modification to tcp's fast recovery algorithm," Tech. Rep., 2012.

[16] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control (bic) for fast long-distance networks," in *Proceedings-IEEE INFOCOM*, vol. 4. IEEE, 2004, pp. 2514–2524.

[17] D. Rossi, C. Testa, S. Valenti, and L. Muscariello, "Ledbat: The new bittorrent congestion control protocol." in *ICCCN*, 2010, pp. 1–6.

[18] S. Abbasloo, Y. Xu, and H. J. Chao, "C2tcp: A flexible cellular tcp to meet stringent delay requirements," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 4, pp. 918–932, 2019.

[19] S. Park, J. Lee, J. Kim, J. Lee, S. Ha, and K. Lee, "Exll: an extremely low-latency congestion control for mobile cellular networks," in *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies*. ACM, 2018, pp. 307–319.

[20] P. Goyal, A. Agarwal, R. Netravali, M. Alizadeh, and H. Balakrishnan, "{ABC}: A simple explicit congestion controller for wireless networks," in *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, 2020, pp. 353–372.

[21] Y. Xie, F. Yi, and K. Jamieson, "Pbe-cc: Congestion control via endpoint-centric, physical-layer bandwidth measurements," in *Proceedings of the 2020 ACM SIGCOMM 2020 Conference*, 2020, pp. 451–464.

[22] F. Lu, H. Du, A. Jain, G. M. Voelker, A. C. Snoeren, and A. Terzis, "Cqic: Revisiting cross-layer congestion control for cellular networks," in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, 2015, pp. 45–50.

[23] K. Winstein and H. Balakrishnan, "Tcp ex machina: Computer-generated congestion control," in *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*. ACM, 2013. [Online]. Available: http://doi.acm.org/10.1145/2486001.2486020

[24] S. Ketabi, M. Buckley, P. Pazhooheshy, F. Farahvash, and Y. Ganjali, "Correlation-aware flow consolidation for load balancing and beyond," *ACM SIGMETRICS Performance Evaluation Review*, vol. 49, no. 4, pp. 105–110, 2022.

[25] P. Pazhooheshy, S. Abbasloo, and Y. Ganjali, "Harnessing ml for network protocol assessment: A congestion control use case," in *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*, 2023, pp. 213–219.

[26] R. Netravali, A. Sivaraman, K. Winstein, S. Das, A. Goyal, and H. Balakrishnan, "Mahimahi: A lightweight toolkit for reproducible web measurement," 2014.

[27] H. Jiang, Y. Wang, K. Lee, and I. Rhee, "Tackling bufferbloat in 3g/4g networks," in *Proceedings of the 2012 ACM conference on Internet measurement conference*. ACM, 2012, pp. 329–342.

[28] A. Narayanan, X. Zhang, R. Zhu, A. Hassan, S. Jin, X. Zhu, X. Zhang, D. Rybkin, Z. Yang, Z. M. Mao *et al.*, "A variegated look at 5g in the wild: performance, power, and qoe implications," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, 2021, pp. 610–625.