DOI 10.5162/iCCC2026/P36

Performance Assessment of 5G for Remote Sensing: KPI Measurements and Practical Insights

Keyvan Shahin, Christian Herglotz, Brandenburgische Technische Universität Cottbus-Senftenberg, Cottbus, Germany
keyvan.shahin@b-tu.de, christian.herglotz@b-tu.de

Abstract

Accurate performance measurements are essential for evaluating how well a 5G network can support practical sensing and data-intensive applications. To support the coverage study, we first developed a measurement framework for collecting key performance indicators (KPIs) of the 5G deployment. The framework, implemented using iperf3, was designed to provide reproducible measurements of throughput, latency, jitter, and retransmissions under a variety of configurations. This foundation enabled the collection of performance data during mobile trials and provided the basis for the coverage mapping and advanced analysis presented in this paper.

1 Introduction

Massive numbers of devices can be supported by fifth-generation (5G) cellular networks, which promise multi-gigabit throughput and sub-10 ms latency [10]. 5G is appealing for data-intensive IoT and sensing applications because of these features. For instance, ubiquitous sensing with low latency and high data rates is essential to environmental monitoring systems and smart city systems. 5G can be used for real-time processing of high-resolution sensor data on edge servers for applications such as unmanned aerial vehicles (UAVs) [1]. Theoretically, 5G's high bandwidth and low latency significantly broaden the range of real-time sensing applications by enabling to perform tasks in which 4G fails, like real-time analytics and 4K/8K video [2].

However, these promises might not be fully realized in actual 5G deployments, particularly in the early releases and implementations. Many operators first implement 5G in non-standalone (NSA) mode on legacy hardware, and coverage is still only partially available. According to [3], new applications will frequently see a mix of 4G and 5G connections. Standard speed tests or simulations may overestimate real-life performance in these circumstances. For IoT/sensor workloads, it is therefore essential to measure end-to-end KPIs under realistic circumstances, particularly throughput and latency. We can only comprehend whether 5G can satisfy the requirements of remote sensing and how deployment factors (location, mobility, and time of day) impact performance with the help of such data.

In this paper, we provide a practical measurement framework and performed an actual field evaluation of these tools for 5G in a campus sensor network setting. Following the methodology explained in [5], we install Raspberry Pi devices with 5G modems and use iperf3 [11] to automate tests [4,5]. After that, we carry out a number of tests to evaluate the uplink and downlink performance of the campus 5G network under actual loads. Quantifying the achievable throughput and latency, identifying variations over time and space, and drawing conclusions for 5G remote sensing applications are the objectives. These basic measurements are primarily performed to validate the functionality of our measurement setup. The main contributions of this paper are:

- Measurement framework: Using Raspberry Pi nodes and iperf3, we create an automated test platform for repeatable 5G throughput and latency measurements.
- Field evaluation: We conduct a study on campus to measure the latency and end-to-end throughput of a commercial 5G deployment under different traffic scenarios similar to that of the Internet of Things.
- Analysis: To highlight the advantages and disadvantages of the 5G network for sensor data, we examine the effects of various factors on the observed KPIs.
- Useful insights: In light of our findings, we provide recommendations for efficient 5G configurations.

This is how the remainder of the paper is structured. The relevant work on 5G KPI measurements and sensing applications is reviewed in Section 2. Our experimental setup and methodology are described in Section 3. The measurement results and discussion are shown in Section 4. A summary of the findings and recommendations for the future are provided in Section 5.

2 Related Work

The goal of recent field research has been to measure the practical performance of 5G networks. One of the first extensive 5G measurements was carried out by [5], who recorded physical-layer metrics and application throughput using commercial smartphones and specialized tools. Although 5G links can reach gigabit speeds under ideal circumstances, they discovered that latency was higher than intended for ultra-low-latency applications and that transport-layer inefficiencies (such as TCP (Transmission Control Protocol)) limited end-to-end utilization. [6] conducted citywide measurements of low- and mid-band 5G vs. 4G in their more recent work. They state that current 5G downlink throughput, with good coverage, is significantly higher than 4G LTE, averaging about 700 Mbps (peaking at about 1 Gbps). The authors point out that larger channel bandwidths, not new modulation or MIMO layers, account for a large portion of this gain.

The authors of [8] focus on methods for extracting KPIs from 5G equipment. After comparing commercial scanners, probe-based tools, and AT-command interfaces, they come to the conclusion that getting fine-grained real-time

KPI data is difficult and frequently necessitates expensive setups. Their research emphasizes the necessity of useful measurement frameworks, such as the one we proposed in this work.

In [7] the authors investigated sensor-equipped UAVs with 5G capability. Their tests demonstrate that uplink latency stays bounded and the application operates as intended when the 5G link is not overloaded. However, pushing the UAV's data rate beyond the 5G capacity caused increased latency, dropped packets, and degraded performance. These studies highlight the importance of throughput and latency as key performance indicators (KPIs) for 5G-based sensing platforms and the necessity of conducting practical tests to determine their limitations.

Finally, a variety of testbeds and tools have been used for 5G experiments. Dedicated campus or private networks allow controlled evaluation: e.g., the Campus5G project deployed a full campus-wide private 5G network (with open RAN (Radio Access Network) architecture) to enable experimental research. Common practice is to use commodity hardware like Raspberry Pis as user equipment, and standard utilities like iperf3 to generate traffic. Such tools provide reproducible measures of throughput and latency (as also noted by industry guides). In our work, we similarly leverage Raspberry Pi endpoints running python scripts using iperf3 library to collect KPI data in a real 5G deployment, combining lessons from these prior testbeds and measurement studies.

3 5G Testbed Configuration and Measurement Methodology

This chapter details our 5G network testbed's hardware and experimental configuration, including the arrangement of the Raspberry Pi devices and the configuration and execution of iperf3-based measurements.

3.1 Hardware Setup and Deployment

Four Raspberry Pi 4 Model B boards were used to construct the testbed, and each board was linked to a Waveshare USB–M.2 (B-key) adapter that was carrying a 5G modem module.

In our configuration, one Pi functioned as the server and three Pis as clients that generated traffic. Our Python scripts, which used the iperf3 library, and Raspberry Pi OS 13 were installed on every device in order to automatically mate traffic tests.

In **Figure 1**, the position of the 5G antenna of the base station and the position of the Pis are illustrated by red and yellow markers, respectively. The client modems function in non-line-of-sight conditions because buildings obstruct the line-of-sight (LoS) between our office and the antenna. This makes an average signal condition to study the 5G network performance. We positioned each USB dongle (with its external antennas) directly outside a window to enhance signal reception. The server was situated approximately two meters away from the three clients which were arranged side by side at one window (**Figure 2**).

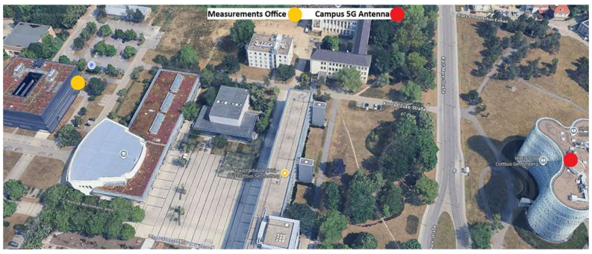

Figure 1 Campus antenna and measurement site

Figure 2 Client 5G dongles setup

3.2 Software Tools and Measurement Scripts

All traffic generation and measurement were done with the iperf3 tool (version 3.x). We developed a multiport server and a client Python script using iperf3 library. Using these scripts, we are able to conduct measurements with desired configurations and log the KPIs throughout these tests.

3.3 Measurement Configuration

Each test run is controlled by a set of input parameters. We repeat each experiment multiple times to gather statistics. The main configuration parameters are:

- Protocol: Either TCP (default) or UDP (User Datagram protocol) for traffic.
- Port: the TCP/UDP port used for iPerf3.
- Duration: 10 (default), seconds per run (the time interval over which iPerf sends data).
- Bandwidth: infinite (uncapped, default) (e.g. 10M for 10 Mbps).
- Parallel Streams: 4 (default), the number of simultaneous iperf3 streams per client).
- Number of Runs: 10 (default), total repetitions of the test, to allow averaging.
- Block Size (TCP window): 131072 (default) bytes, this is the iPerf3 default buffer size.

These configuration options, which are constant throughout each test, are set in the Python scripts. To investigate their impact on performance metrics, we changed one or more of these configurations to make various test scenarios.

3.4 Performance Metrics

From each iperf3 test we log the following KPIs (as reported by iperf3) along with a timestamp:

- Bytes Sent: Total data transferred by the client.
- Throughput (Mbps): Achieved data rate in megabits/second.
- RTT (ms): Round-Trip Time measured by iperf3 on the TCP connection (computed as the average TCP ACK latency).
- Retransmits: Number of TCP retransmitted segments (for TCP tests only).
- Jitter (ms): Packet delay variation (measured during UDP tests).
- Packet Loss (%): UDP packet loss percentage (UDP tests only).
- CPU Utilization (%): The operating system CPU load on each Raspberry Pi (logged via a system call on the Pi).

Additionally, we timestamp each run to later correlate with the logs from other server/clients.

3.5 CPU Usage and Power Considerations

During tests, we kept an eye on each Pi's CPU utilization as an indicator of the power consumption for Raspberry Pi devices [9].
Overall, by monitoring traffic parameters and observing how they affect throughput, latency, and other performance metrics, this testbed and measurement methodology enable us to systematically study 5G network KPIs. To isolate the effects of each setting on the recorded KPIs, the above detailed input configurations were changed across scenarios.

4 Measurements and Results

We carried out controlled experiments with different channel bandwidth, bulk transfer size, transmission duration, and client concurrency to investigate how configuration decisions impact KPI behavior.

Figure 3 Impact of Bandwidth on throughput

First, we discuss the networks performance in terms of throughput. Network performance was clearly impacted by changing the configured bandwidth. Restrictive bandwidth values resulted in lower throughput (**Figure 3**) and slightly higher latency (**Figure 4**). These patterns demonstrate that both achievable data rate and latency are affected by available bandwidth.

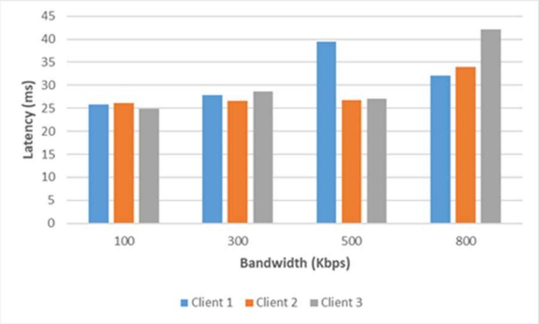

Figure 4 Impact of Bandwidth on latency

Figure 5 Impact of Bulk Size on Throughput

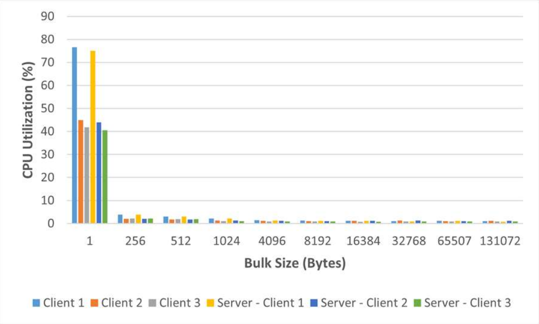

Figure 6 Impact of Bulk Size on CPU Utilization

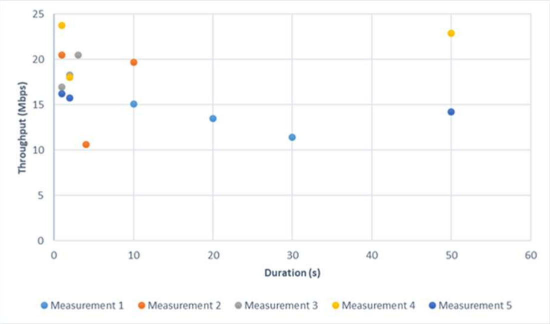

Figure 7 Impact of Transmission Duration on Throughput

DOI 10.5162/iCCC2026/P36

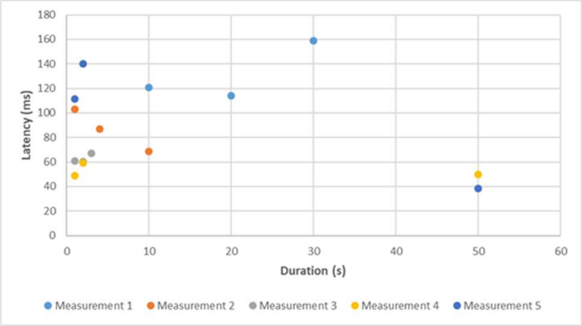

Figure 8 Impact of Transmission Duration on Latency

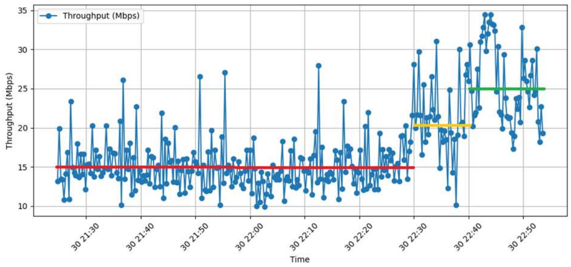

Figure 9 Impact of Interference on Throughput

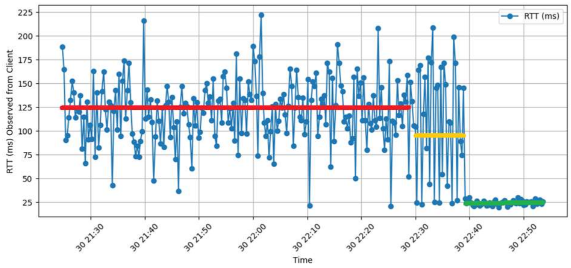

Figure 10 Impact of Interference on Latency

Performance was not noticeably impacted by bulk transfer size unless the bulk size was set too low (**Figure 5**). While values over 256 Bytes contributed to low CPU utilization, very small bulks introduce very high CPU utilization (**Figure 6**).

Figure 7 and **Figure 8** indicate that the duration of transmission seems to have no meaningful impact on the throughput and the latency.

We also ran multiple clients with different end times to investigate interference. As each client's task was finished, we clearly observed an increase in the throughput and a decrease in the latency (round-trip time) of the other working clients. We are demonstrating the throughput and latency of the last client to show how they change when other clients' tasks are finished (**Figure 9**,**Figure 10**).

We used UDP to repeat the majority of the tests, but most UDP runs failed or timed out in our non-line-of-sight weak-signal environment. Additionally, there were significant differences in CPU usage: TCP rarely exceeded ~10%, while UDP saturated almost 100% of the server CPU. Therefore, TCP proved to be more reliable and much less demanding for our sensing platform, even though UDP can theoretically deliver higher throughput.

5 Conclusion and Future Work

In this work, we developed an automated tool to measure 5G network KPIs under various configurations and real-world scenarios. We then share some of our findings, considering the impact of those configurations on the KPIs here.

As a continuation of this work, the developed tools and methods are being used in conjunction with drones and land vehicles to create a 5G network coverage map of the campus.

6 References

[1] K. Shahin et al., "A Modular Communication Architecture for Adaptive UAV Swarms," *2023 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, Berlin, Germany, 2023, pp. 1-6, doi: 10.1109/COINS57856.2023.10189245.

[2] C. Herglotz et al., "Designing Intelligent Sensor Networks: A Comprehensive Survey of Sensing, Edge AI, and 5G+/6G Connectivity," in IEEE Access, vol. 13, pp. 172306-172325, 2025, doi: 10.1109/ACCESS.2025.3615205.

[3] Banerjee, A. et al. (2024), "5G enabled smart cities: A real-world evaluation and analysis of 5G using a pilot smart city application," Internet of Things, vol. 28, p. 101326.

[4] Gkagkas, G. et al. (2025), "The Behavior of an IoT Sensor Monitoring System Using a 5G Network and Its Challenges in 6G Networking," Electronics, vol. 14, no. 16, p. 3167.

[5] Xu, D. et al. (2020), "Understanding Operational 5G: A First Measurement Study on Its Coverage, Performance and Energy Consumption," Proc. ACM SIGCOMM '20. Pages 479 – 494 https://doi.org/10.1145/3387514.3405882

[6] Rochman, M. I. et al. (2024), "A Comprehensive Real-World Evaluation of 5G Improvements over 4G in Low- and Mid-Bands," arXiv preprint arXiv:2312.00957.

[7] Damigos, G. et al. (2023), "Performance of Sensor Data Process Offloading on 5G-Enabled UAVs," Sensors, vol. 23, no. 2, p. 864.

[8] Kesir, S. et al. (2024), "A Comparative and Measurement-Based Study on Real-Time Network KPI Extraction Methods for 5G and Beyond Applications," arXiv preprint arXiv:2504.16039.

[9] Kesrouani et al., A Preliminary Study of the Energy Impact of Software in Raspberry Pi devices. 29th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2020, Bayonne, France. hal-02936861

[10] Attar H et al., 5G System Overview for Ongoing Smart Applications: Structure, Requirements, and Specifications. Comput Intell Neurosci. 2022 Oct 11;2022:2476841. doi: 10.1155/2022/2476841. PMID: 36268153; PMCID: PMC9578857.

[11] https://iperf.fr/iperf-doc.php