An Analysis of 5G Energy Consumption on the Raspberry Pi 5

Claire Lu, Anthony Padilla

[1] University of California, Riverside

Abstract **:**

The development and increasing commercial accessibility of Fifth Generation (5G) technology demonstrates a significant improvement over previous mobile wireless network generations. While it boasts improvements in terms of lower latency, increased throughput, and interconnection, a major weakness of the technology is its high energy cost. To study and analyze the causes of this cost, we approached data collection through an unorthodox method, that being the use of 5G on Raspberry Pi 5. The purpose of utilizing this device to study 5G data is to provide a greater degree of control when isolating parameters and lower level information, rather than using proprietary hardware. This paper details our implementation of the test environment and our subsequent results.

1 Introduction

After a few years having passed since the deployment of 5G, it is more than safe to say that the technology is invaluable. Being a massive improvement over previous generations in every way, 5G has become the new standard in wireless technology. However, it is not without its drawbacks. While 5G provides lower latency, and faster download speeds, the technology is still held back by its immense power consumption. This weakness is holding back its application in other types of devices, such as drones or robotics. With 6G on the horizon, it is surprising that 5G has still not been heavily researched in such as a fashion. Our goal is to hopefully gain insight on the causes of this weakness, by taking an unorthodox approach, and working with 5G on a non-cellular device.

If we were to perform this measurement study using proprietary devices, it is possible that we would be limited in the type of information we could extract. Being able to achieve root access, and monitor accurate power consumption, for example, might not be possible, or prove to be notably difficult. Establishing the testbed on a Raspberry Pi 5 would provide us more control over all aspects of the device, including what applications are running in the background, and being able to observe, without restriction, what is occurring on the machine at all times. There are also benefits in data collection automation, provided by working with an open-source Linux distribution such as Raspian.

2 Background

Fifth Generation Mobile Wireless Networks, 5G, improve on performance metrics significantly as compared to previous generations, due to across the board advancements and new technologies, (For example carrier aggregation, usage of mmWave frequency bands). These technologies enable higher data rates, increased download speed, and low latency. The added complexity, however, incurs tradeoffs in energy demand greater than its 4G predecessor. Compared to previous generations, 5G contributes to higher power consumption. This is the result of denser deployments of base stations for coverage, along with complex signal processing required by massive MIMO. Critically, the relationship between throughput and power consumption needs to be understood, as energy efficiency is becoming an important concern for mobile networks.

This weakness of 5G was primarily discussed in Xu et al., 2020, one of the first measurement studies on 5G. Their study shows the significant energy cost of 5G, as compared to previous generations. They note how, when performing their experiments, the 5G module accounted for 55% of the average energy cost when running any given application. Being on of the first studies, it leaves room for future work, and at that time, suggested the best solution would be a dynamic fallback to 4G when necessary, only utilizing 5G technology when necessary.

Prior measurement studies, such as the aforementioned one, rely on commercial mobile devices, utilizing proprietary hardware that could limit obtainable diagnostic data. Thus, we will experiment on a testbed that allows for greater control in measurements and data collection of low-level interactions. In this work, we present a controlled 5G testbed built on a Raspberry Pi 5 equipped with a commercial 5G HAT. By using this experimental model, we are able to collect more diagnostic data and isolate experimental conditions to directly study power consumption.

3 Methods

Our final established testbed made use of a Rasberry Pi 5, and the RM530N-GL 5G HAT+ from Waveshare (Waveshare, n.d.). We used a prepaid Mint Mobile^{TM} SIM card with 5G functionality, as attempting to use the open source srsRAN project did not work with our module. Waveshare provided a power monitoring script, named **INA219.py**, which output current power statistics to a terminal. This was edited to produce **ucrINA.py**, which creates a .csv file containing power consumption information rather than being a live visual.

We developed two programs, named **DataLoggr** and **Zippr**. The former was a program that captured diagnostic messages, interface traffic, and power consumption for a given test, while the latter visualized throughput and power consumption.

We performed 4 different experiments to observe and test our experimental setup. We tested:

- *pingTest*: Pinging google.com, with random file downloads in between
- *spotify*: Streaming music from Spotify
- *breakingbad*: Streaming a show from Netflix
- *squirrel*: Streaming a livestream from YouTube

3.1 pingTest

For *pingTest*, we pinged www.google.com for a 2 minute duration. In between the pinging, an operation of requiring little throughput, we performed a download of a 100mb .zip file 4 times. The motivation for this was due to a constant pinging, the module was unlikley to enter sleep mode, and observing the changes from low to high throughput could be insightful. [.zip files were downloaded from thinkbroadband.com]

3.2 *spotify*

The *spotify* test was 4 minutes of interacting with the Spotify web client, interacting with it as a normal user. This included querying songs, skipping forward in succession, and simply letting the music play. As compare to a live video, or a streaming a movie, audio tends to be of a lower size, and could perhaps be loaded into a buffer rather quickly. It is possible that it could be one of the least intensive of the streaming tests for this reason.

3.3 *breakingbad*

This test included loading and streaming a Netflix Show, breaking bad. The test mainly tested loading and watching straight from a link, so this does not include navigating the app. As compared to *spotify*, video and audio streaming requires more throughput, being of a larger size. However, still compared to a live stream, the use of buffers is still possible as it is not a live streaming of data.

3.4 *squirrel*

The final test, *squirrel*, consisted of streaming a YouTube live stream, which we chose to be a 24/7 video feed of squirrels and birds. Notably different than previous tests, it is the only one that needs to updated live. There isn't an ability to load data into a buffer, as it doesn't necessarily exist. This could perhaps impact the module consisting sleeping due to waiting for more data, increasing power costs.

4 Results

After running the 4 experiments with the mentioned **DataLoggr**, the graphs were produced using **Zippr**, ensuring aligned throughput and power times. Below are the visuals produced:

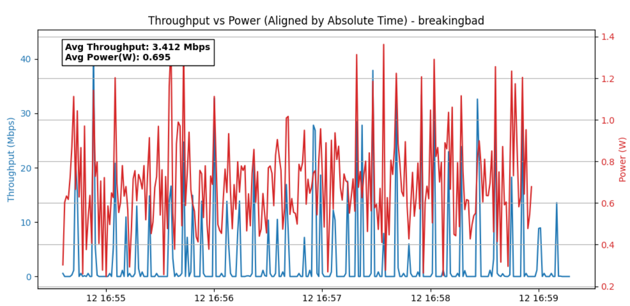

Figure 1. Graph for *breakingbad*. Being a stream of a show, we see that indeed there is consistently high throughput due constant filling of the buffer, with both audio and video.

5 Discussion

After visualizing the experiments, we can see that some of the theories we had regarding behavior held true. Comparing *squirrel*

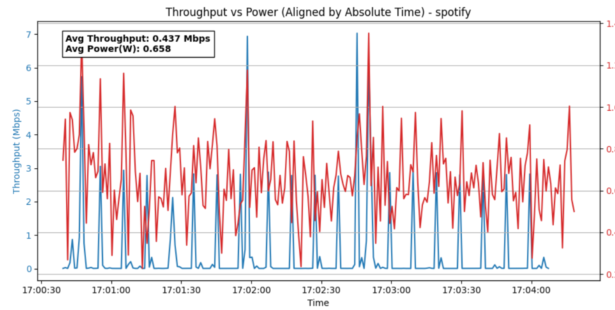

Figure 2. In contrast to **Figure 1**, *spotify* was of lower throughput, with about 0.5(W) less average power consumption.

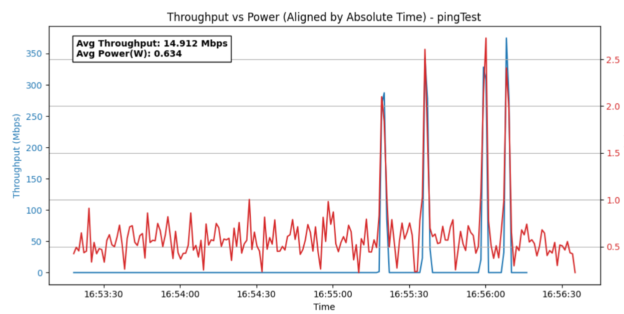

Figure 3. Graph for **pingTest**. Average throughput was near zero in terms of Mbps, and we can see the 4 peaks corresponding to the 100mb file downloads.

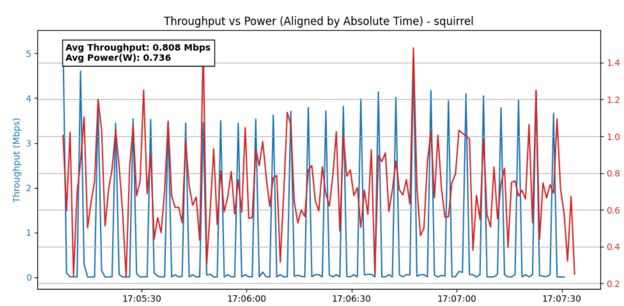

Figure 4. *squirrel* visualization. The most telling of the visualizations, we see consistent peaks and dips in throughput, almost periodic.

with the other experiments, we see that it had the highest average power consumption, despite now having the highest average throughput. This could be due to the constant dead periods, waiting for updates coming from the livestream. Parsing the diagnostic messages associated with this test did show CDRX message logs, which relate to sleep functionality.

Another notable observation is that of *pingTest* and the two other streaming experiments, *spotify* and *breakingbad.* The throughput metric can be misleading, as the large file downloads were outliers for the overall test. The more notable fact is that despite having almost constant close to zero throughput, it had similar power consumption to the other tests. While there could be other factors, such as the other experiments having true 0mbs values, having constant data to be able to grab could prevent the HAT from sleeping, as discussed with the previous test.

Note, in order to perform said tests, we needed to provide two 5V power sources, one for both the Pi and the 5G module. The 5G module would experience a fatal shutdown if the requirement wasn't meant, showing high cost even during testbed setup.

6 Future Work

Prior work has investigated how cell selection and mobility management influence network performance and user experience. From Q Li, 2021, we can see that standard cell selection leads to a suboptimal throughput. Reconfiguration and tuning the cell selection has improved effects on network efficiency. Cell selection and configurations affect the throughput and subsequently, energy consumption. Noting that there were significant improvements in throughput after optimizing cell selection parameters, it should also be possible to use the same methodology to optimize 5G energy cost based on selected cells.

In our current testbed, we are able to access accurate power consumption information, as well as throughput and diagnostic messages. We are able to access cell configuration information, and the next step is to now control these configurations. Making the testbed mobile, and testing separate configurations could reveal what parameters have the highest influence on energy cost, and could allow us to configure these protocols to reduce, similarly to how Q Li, 2021 improved upon throughput.

7 Conclusion

While the tests done were not fully extensive, the current experimental testbed and performed experiments show the promising proof of concept as an insightful and valuable testing setup. Direct access to power consumption via the INA module on the 5G HAT, as well as the data collection possibilities due to working with a Linux distribution, allowed us to bypass common restrictions found when working with a mobile device. With our proof-of-concept realized, the next steps involve making the experiment setup mobile, and further testing different variables that can contribute to high energy consumption.

Acknowledgements

This research was overseen by Professor Zhaowei Tan, assisted by graduate student Zhutian Liu. This research received funding support from CRA UR2PhD program, as part of the Summer 2025 REU.

References

Q Li, C Peng (2021). "Reconfiguring Cell Selection in 4G/5G Networks". In: *N/A*. Accessed: 2025-08-21. DOI: 10.1109/ICNP52217.2021.00015. URL: https://icnp21.cs.ucr.edu/papers/icnp21camera-paper36.pdf.

Waveshare (n.d.). *RM530N-GL 5G HAT+*. Accessed: 2025-08-21.

Xu, Dongzhu et al. (2020). "Understanding Operational 5G: A First Measurement Study on Its Coverage, Performance and Energy Consumption". In: *Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication (SIGCOMM '20)*. DOI: meow. URL: https://xyzhang.ucsd.edu/papers/DXu_SIGCOMM20_5Gmeasure.pdf.

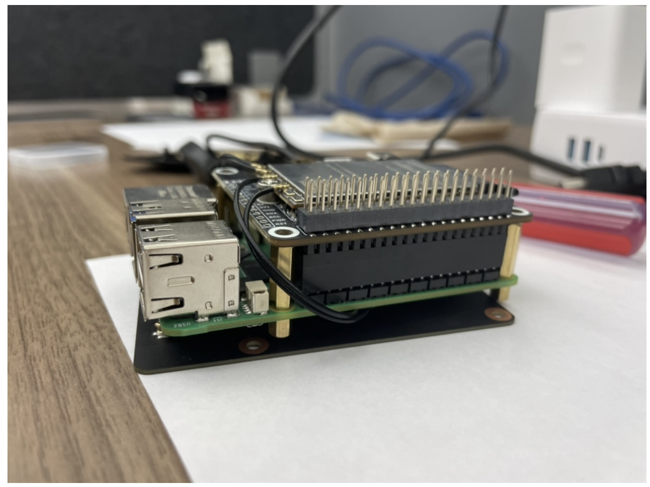

Figure 5. Raspberry Pi 5 with RM530N-GL 5G HAT+

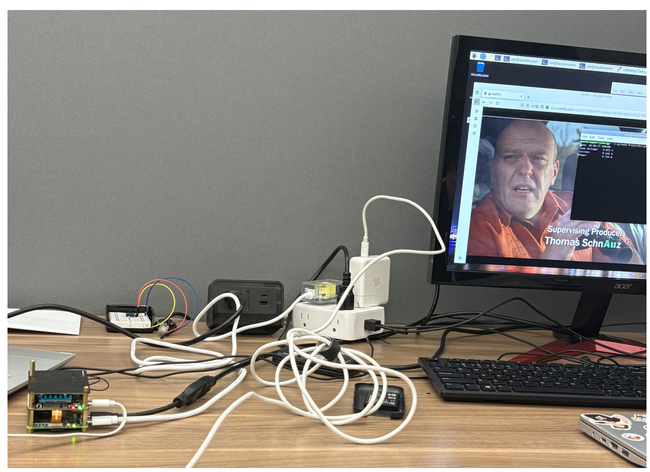

Figure 6. Current testbed setup, with two power sources, separate machine for diagnostic monitoring, and power tracing (for *breakingbad*