**Latency analysis**

**Objective of this proposed work:**

In this work we intend to analyze the latency of two different networks, (Verizon and Airspan) as well as three different User Equipment Devices (M1000 Inseego Hotspot, Airspan AirSpot, and Peplink). The M1000 only works with the Verizon Network and the Airspan Airspot only works with the Airspan network, whereas the Peplink works with both networks but with LTE only on the Verizon network, since it does not support the 28GHz mmWave 5G of the Verizon network. There are three different states in which these devices operate, namely the inactive, active and the idle state. The time taken by the device to reach the active state and slip back to the inactive state is considered and studied exclusively in this work for a better understanding of the 5G operations.

**Evaluation tested and process involved:**

The evaluation testbed comprises of two virtual machines, namely the EdgeCompute and distributed energy resource (DER). The EdgeCompute is connected to the LabNet while the DER is connected to either Verizon or Airspan depending on the requirements. The latency information is collected by running a script file on the DER.

The script file comprises of the ping command, IP address of the EdgeCompute VM, timeout command, and the total number of iterations. The latency data associated with each of these iterations will be stored independently as a CSV file. The CSV files are processed, and the Python was utilized for plotting these graphs. Here, a timeout period of 100 and 50 different iterations were considered. The script file was executed on the DER. The latency values for three different sleep times, namely 1s, 10s and 25s, were collected in this work. The reason for choosing these three values is that we think act as a threshold for the system to slip from active to idle and to inactive state respectively.

**NOTE:**

1. Run number – Each run is composed of a variable *timeout period* before a run followed by 100 pings once per second. The Figure 1 accurately summarizes the run number for better understanding. As shown in Figure 1 the first iteration is removed, this is because there is no way to know how much time had passed between the last message sent across the 5G network and the first ping of that iteration. By removing the first iteration we can control the amount of time that passes (Variable Timeout Period) between messages going over the 5G network, allowing us to control which state the User Equipment is in before sending a message.

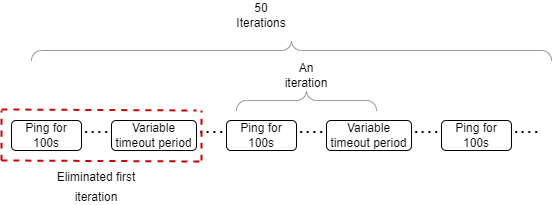


Figure 1 Illustration of a run

1. The 5G devices have three RRC States, namely *Active*, *Inactive*, and *Idle*. From empirical data analysis based on the latencies of the first message after a Variable Time Out Period the latencies change respectively to the RRC State of the User Equipment with latency thresholds of < 100ms (*Active State*), 150-450ms (Ina*ctive State*), and > 500ms (*Idle State*). With this assumption, we analyze the following graphs.
2. Violin plots were utilized to plot the graphs as they present the accurate description of the collected latency data. The white dots present the mean of the collected latency.
3. The raw data utilized for the plotting is labelled as follows:

|  |  |
| --- | --- |
| **Sleep time** | **Name of the folder** |
| 1s | 1s-delay-13th |
| 10s | 10s-delay-13th |
| 25s | 25s-delay |

1. The files in each of these folders are labelled in a specific manner: <iteration-number>.<name of the file>.<time-stamp>
2. Python script file that was utilized to generate the plots is included under name “Latency-analysis-different-delays.py”

**Statistical representation of the latency plots**

1. Sleep time of 1s

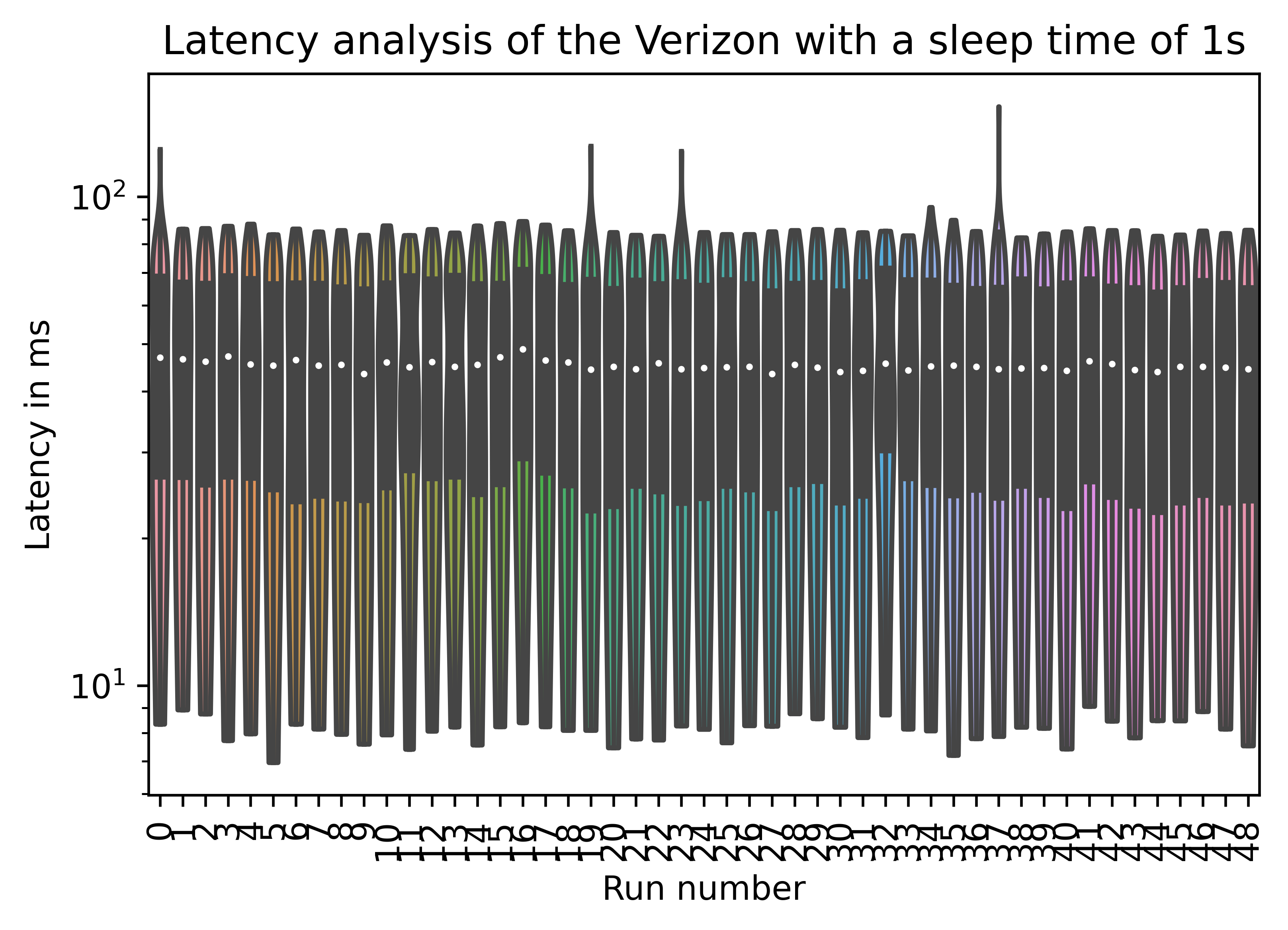


Figure 2 Latency analysis of sleep time of 1s

1. 10s – delay

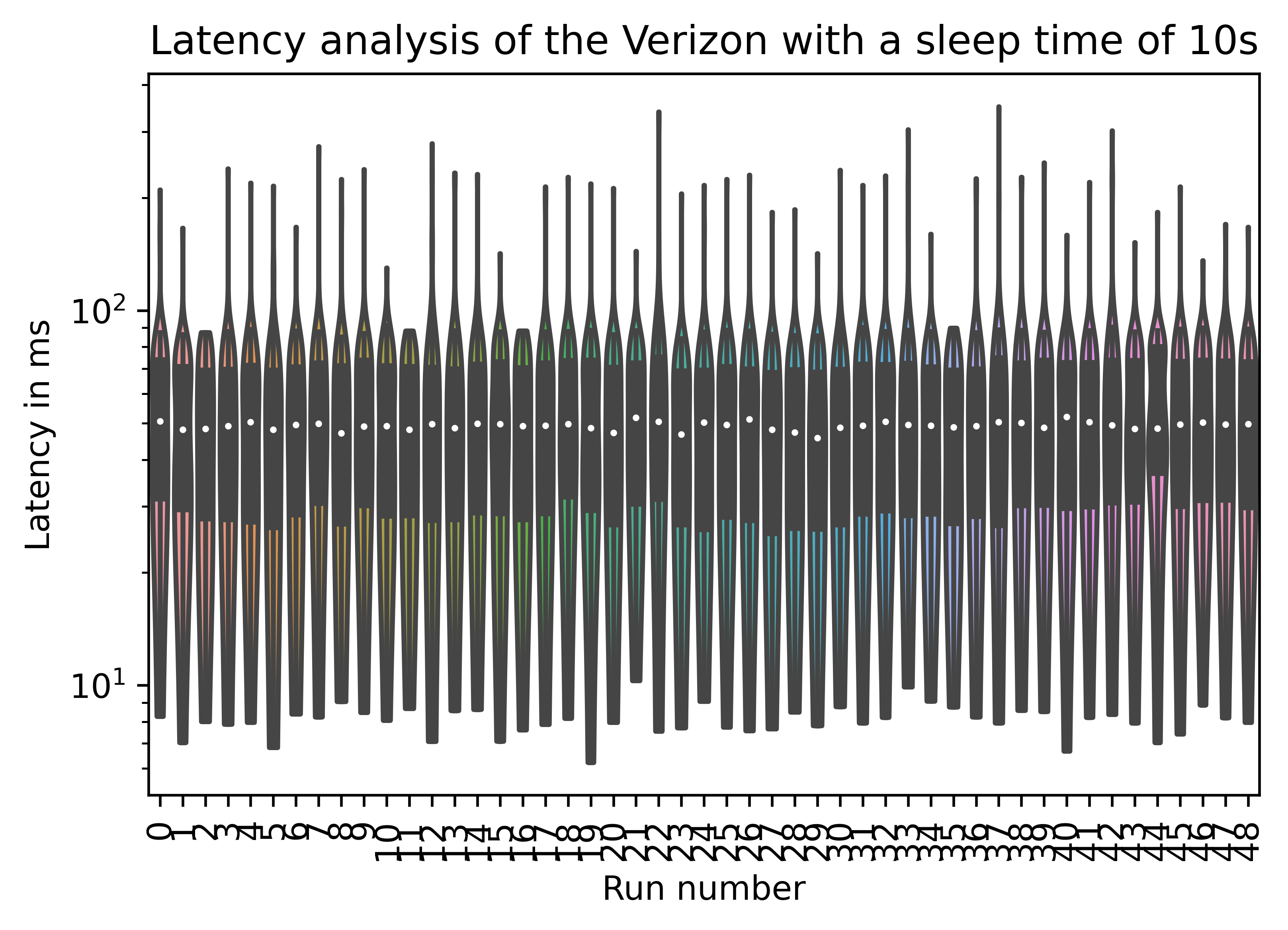


Figure 3 Latency analysis of sleep time of 10s

1. 25s – delay

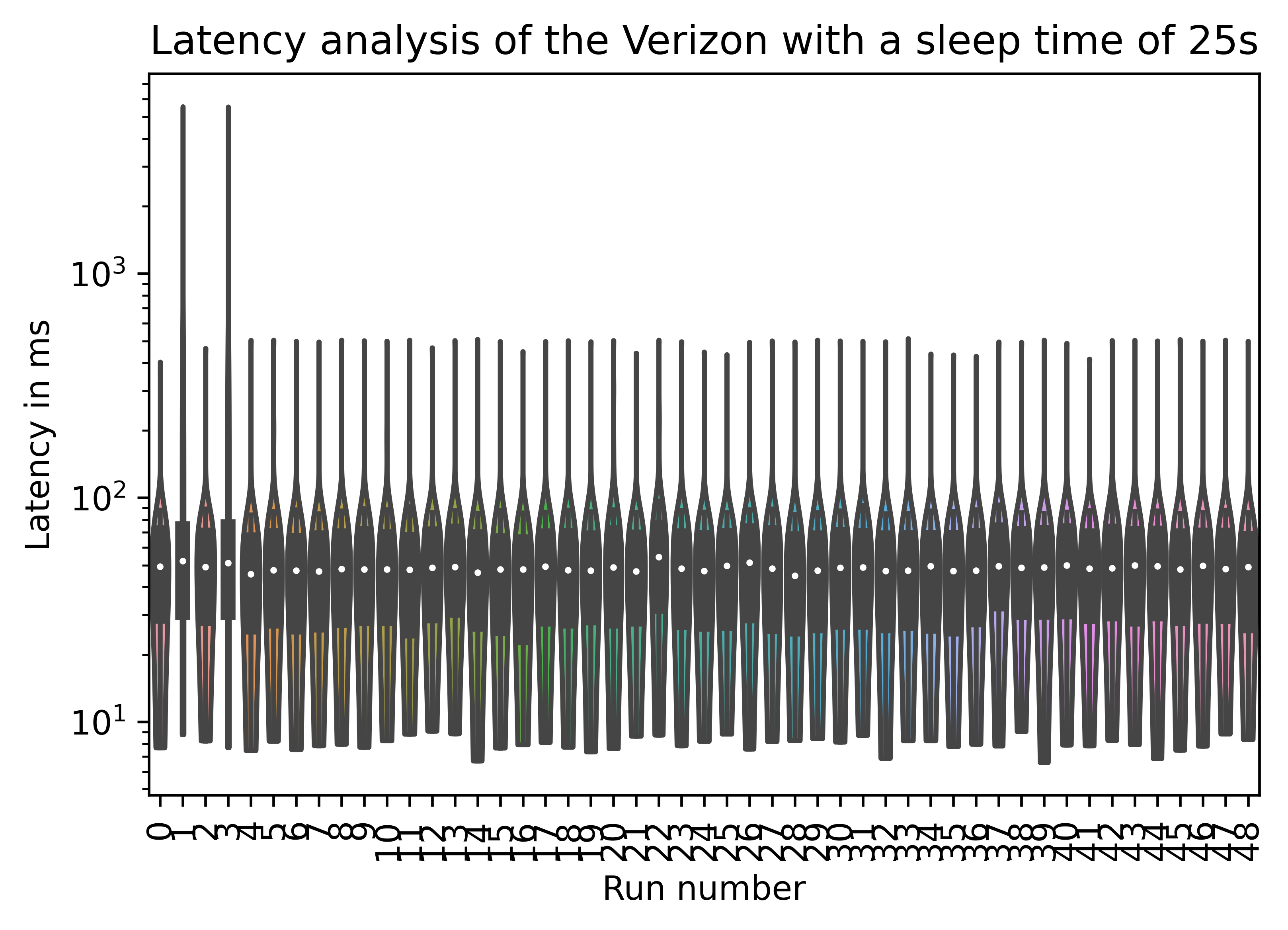


Figure 4 Latency analysis of sleep time of 25s

**Observations:**

1. The sleep time plays a significant role in transitioning the device between the three different states – active, inactive and idle. By understanding the violin plot structure, it is easy to understand the latency pattern.

* The violin plots have three main components – the outliers (represented by the thin top line) which when there is an outlier it is typically the first message that is sent in that iteration, the median (represented by the white dot) and the inter-quartile range (represented by the colored box).
* The wider and the narrower region of this density plots represents the higher and lower occurrence of the values respectively.

1. Let us consider the sleep time of 1s (presented in Figure 2). The network is in active state as its latency value lie in between 10-100ms for most times. However, in run numbers 0, 20, 22, 24, and 37, we believe the first value (or the highest peak also referred to as outliers) depicts the changing of the system to inactive state from active state even though there is only a 1 second delay between messages. It could also be the case that there is just increased latency due to other factors at this time.
2. In Figure 3, we can assume that network changes into inactive state for almost all the run numbers with a sleep time of 10s. The first message lies well above 100ms for which we believe it indicates that the system is in inactive state due to the extra time required to change from inactive to active states which requires less time than changing from idle to active states.
3. In Figure 4, we can assume that network changes into the idle state for all the run numbers with a sleep time of 25s. The first message lies well above 400ms which we believe that indicates the system is in idle state due to the extra time required to change from idle to active states.
4. Currently the latency added due to state changes is assumed based on empirical analysis of the behavior of the devices and finding that there are commonly three different latency ranges based on how much time passes between two messages. Correlating this with the fact that there are three different RRC states is how we reached this conclusion. Future testing is planned to use the WaveJudge 5000 to analyze the over the air messages between the User Equipment Devices and the Cell Site which allows us to see if and when the different RRC states change.