RRC State Transition Simulation: Idle, Connected, and Inactive Modes in 5G Networks

Dr. K. A. Nethravathi[#], Harshit Thakkar, Shafiya K M*, Junaid Ahmed**

[#]Assistant Professor, Department of Electronics and Communication, RV College of Engineering, Bengaluru-560059

**Student, Department of Electronics and Communication, RV College of Engineering, Bengaluru-560059*

Abstract— **The introduction of the Radio Resource Control (RRC) Inactive state in 5G New Radio (NR) networks represents a significant step toward improving energy efficiency while maintaining low latency. Unlike the traditional LTE model that relies solely on Idle and Connected states, the RRC Inactive state allows user equipment to temporarily suspend data transmission while preserving connection context [1]. However, the effectiveness of this state is strongly influenced by the configuration of inactivity timers, which are often statically defined and unable to adapt to varying traffic conditions. In this work, a real-time simulation framework is developed to analyze RRC state transitions among Idle, Connected, and Inactive modes using a model-based design approach in MATLAB and Simulink. A novel Adaptive Inactivity Timer (AIT) mechanism is proposed, which dynamically adjusts transition thresholds based on different traffic profiles such as bursty IoT communication and continuous data streaming [2]. The system is implemented using State flow to ensure strict adherence to 3GPP specifications and is integrated with an interactive graphical interface for real-time control and visualization. Simulation results demonstrate that the proposed adaptive approach significantly reduces energy consumption compared to conventional static timers, while maintaining low latency and minimizing signaling overhead. The findings highlight the importance of intelligent RRC state management and validate the RRC Inactive state as a key enabler for energy-efficient 5G networks, particularly for IoT and massive machine-type communication scenarios [4].**

I. Introduction

The rapid evolution of mobile communication systems has transformed cellular networks from simple voice-centric platforms into highly flexible infrastructures capable of supporting a wide range of services. Fifth Generation (5G) New Radio (NR) networks are designed to accommodate diverse and often conflicting requirements, including enhanced mobile broadband (eMBB), ultra-reliable low-latency communications (URLLC), and massive machine-type communications (mMTC). While these use cases demand high data rates, low latency, and massive connectivity, they also introduce new challenges related to energy efficiency, signaling overhead, and network scalability [5].

One of the key contributors to energy consumption in cellular devices is the Radio Resource Control (RRC) mechanism, which governs how user equipment (UE) connects to and communicates with the radio access network (RAN). In Long Term Evolution (LTE) systems, the RRC state machine consists primarily of two states: Idle and Connected. In the Idle state, the UE consumes minimal power but must undergo a full connection setup procedure before transmitting or receiving data. In contrast, the Connected state allows active data exchange with low latency but at the cost of significantly higher power consumption due to continuous monitoring of control channels and active radio resources [3].

This binary state model, while effective for traditional human-centric traffic, becomes inefficient in modern 5G scenarios. Applications such as Internet of Things (IoT) sensing, smart meters, and periodic telemetry generate short, sporadic bursts of data separated by long idle intervals. For such traffic patterns, frequent transitions between Idle and Connected states result in excessive signaling, increased latency, and unnecessary energy drain. Even after data transmission is completed, the UE often remains in the Connected state for a predefined inactivity period, leading to what is commonly referred to as the “radio tail” energy waste problem [6].

To address these limitations, 5G NR introduces a new intermediate RRC state known as the RRC Inactive state [8]. This state is designed to combine the low energy consumption characteristics of the Idle state with the fast resume capability of the Connected state. In the Inactive state, the UE releases dedicated radio resources while retaining essential connection context within the RAN. As a result, when data transmission needs to resume, the UE can quickly transition back to the Connected state using a lightweight resume procedure rather than performing a full connection setup. This design significantly reduces signaling overhead and access latency, particularly for devices with intermittent traffic.

Modern wireless communication systems have evolved far beyond their original purpose of enabling basic voice communication. Today’s cellular networks are expected to seamlessly support high-definition video streaming, real-time industrial control, autonomous systems, and billions of low-power devices operating simultaneously. The Fifth Generation (5G) New Radio (NR) standard was developed to address this wide spectrum of requirements by introducing flexible architectural and protocol enhancements at both the radio and core network levels. However, achieving this flexibility comes at the cost of increased system complexity, particularly in the way devices manage their connection states and energy usage.

From the perspective of user equipment (UE), energy efficiency has become a critical design consideration. Many 5G-enabled devices, especially sensors and embedded systems, are expected to operate for extended periods on limited power sources such as batteries or energy-harvesting modules. In such scenarios, even small inefficiencies in radio operation can significantly reduce device lifetime. A substantial portion of this energy consumption is influenced by how the UE transitions between different operational states defined by the Radio Resource Control (RRC) protocol.

The RRC layer plays a central role in coordinating communication between the UE and the radio access network. It determines when the device should actively transmit data, when it should listen for network messages, and when it can safely enter low-power modes. While earlier cellular generations relied on relatively simple state models, the traffic patterns observed in modern networks are far more diverse. Continuous data streams coexist with sporadic, event-driven transmissions, placing conflicting demands on latency, responsiveness, and power consumption.

In real-world deployments, many applications generate data only occasionally, yet they require fast network access whenever communication is needed. Examples include environmental monitoring, smart city infrastructure, wearable health devices, and industrial automation systems. For these applications, repeatedly establishing and releasing full network connections introduces unnecessary signaling overhead and delays, while keeping the device fully connected wastes energy during inactive periods. This mismatch between application behavior and traditional connection management strategies highlights the need for more intelligent RRC state handling.

The introduction of an intermediate operational state in 5G NR represents a significant step toward addressing this challenge. By allowing devices to retain essential connection information while releasing active radio resources, the network can offer a balance between responsiveness and energy efficiency. However, the effectiveness of this approach depends strongly on how transition decisions are made, particularly with respect to inactivity durations. Fixed configuration values may work well for one type of traffic but perform poorly for another, leading to inefficient use of both device and network resources.

Given these challenges, there is a clear need for practical tools that allow researchers and engineers to observe, analyze, and evaluate RRC behavior under realistic traffic conditions. Simulation-based approaches provide a controlled environment for studying the impact of different design choices without the cost and complexity of full-scale network deployment [6].At the same time, such simulations must remain faithful to standardized behavior to ensure that conclusions are meaningful and transferable to real systems.

This project addresses these needs by developing a real-time RRC state transition simulation framework for 5G NR systems. The framework models the interaction between user traffic, inactivity timers, and RRC state changes in a time-accurate manner, enabling detailed observation of system dynamics. By incorporating adaptive mechanisms that respond to observed traffic characteristics, the simulator demonstrates how intelligent timer management can reduce unnecessary energy expenditure while preserving the responsiveness required by modern applications.

Through this work, the project aims to provide deeper insight into the practical behavior of RRC state machines and to highlight the importance of adaptive control strategies in next-generation wireless networks. The proposed framework serves not only as an analytical tool but also as an educational platform for understanding the trade-offs involved in 5G connection management.

Despite its advantages, the performance of the RRC Inactive state is highly dependent on the configuration of inactivity timers that control transitions between Connected and Inactive modes [9]. In many existing deployments and simulation studies, these timers are statically configured with fixed durations. While static timers simplify implementation, they fail to account for the dynamic nature of traffic patterns across different applications. A timer value that is suitable for continuous video streaming may be highly inefficient for bursty IoT traffic, and vice versa. Consequently, static timer configurations often lead to suboptimal energy efficiency and increased latency in real-world scenarios [6].

This paper presents a real-time simulation framework for analysing RRC state transitions in 5G NR, focusing on Idle, Connected, and Inactive states. The proposed framework is implemented using Simulink and State flow to ensure strict adherence to the transition rules defined in 3GPP standards. A key contribution of this work is the introduction of an Adaptive Inactivity Timer (AIT) mechanism that dynamically adjusts state transition thresholds based on observed traffic profiles. Unlike conventional static timers, the proposed approach enables the system to adapt to both bursty and continuous traffic patterns, thereby reducing unnecessary energy consumption while maintaining low latency [10].

II. OBJECTIVES

- To model the complete 5G RRC state machine: Develop a structured and standards-compliant RRC state machine consisting of Idle, Connected, and Inactive states using a model-based design approach [2]. The objective is to closely follow the transition logic specified by 3GPP standards, ensuring that the simulation reflects realistic network behavior rather than simplified approximations.

- To implement realistic RRC state transition triggers: Simulate key network events that influence RRC state transitions, including uplink data transmission requests, inactivity timer expirations, and paging signals from the network. This objective ensures that the system responds dynamically to real-time events in a manner consistent with practical 5G deployments [6].

- To design and integrate an Adaptive Inactivity Timer (AIT) mechanism: Move beyond conventional static timer configurations by developing an adaptive timer that dynamically adjusts transition thresholds between

Connected and Inactive states based on observed traffic patterns. This objective aims to address the inefficiencies of one-size-fits-all timer values, particularly in scenarios involving heterogeneous traffic profiles [12].

- To evaluate performance across different traffic profiles: Analyze the behavior of the RRC state machine under varying traffic conditions, such as bursty IoT communication and continuous data streaming [3]. The goal is to understand how adaptive timer logic impacts energy consumption, latency, and state transition frequency for different types of applications.

- To quantify energy consumption in real time: Incorporate an energy calculation subsystem directly within the simulation loop to continuously monitor and accumulate power usage across different RRC states. This objective enables a direct and transparent comparison between static and adaptive state management strategies without relying on post-simulation estimations [15].

- To minimize signaling overhead while maintaining low latency: Investigate how efficient utilization of the RRC Inactive state and adaptive timers can reduce unnecessary signaling caused by frequent connection setups and releases, while still ensuring fast transition back to the Connected state when data transmission is required.

- To develop an interactive visualization and control interface: Create a user-friendly graphical interface that allows manual triggering of network events and real-time visualization of RRC state transitions [9]. This objective enhances interpretability by making the otherwise invisible protocol behavior observable and intuitive.

- To validate the practical relevance of the RRC Inactive state: Demonstrate, through simulation results, that the RRC Inactive state serves as an effective compromise between energy efficiency and responsiveness, particularly for IoT and mMTC use cases that dominate modern 5G traffic.

III. METHODOLOGY

This section describes the design approach, implementation strategy, and evaluation methodology adopted to simulate and analyze Radio Resource Control (RRC) state transitions in a 5G New Radio (NR) environment. The proposed methodology emphasizes realism, standards compliance, and real-time observability, ensuring that the developed framework accurately captures the dynamic behavior of the RRC state machine while allowing meaningful performance evaluation [1].

A. Model-Based Design Approach:

A model-based design methodology was adopted to systematically represent the discrete and event-driven nature of the RRC protocol. Unlike script-based simulations, which often approximate protocol behaviour, model-based design enables explicit definition of states, transitions, and triggering conditions. MATLAB Simulink was chosen as the primary simulation environment due to its strong support for system-level modelling and seamless integration with State flow for finite state machine implementation.

RRC State Transition Model:

$$x(t+1) = \delta(x(t), u(t))$$

This equation forms the backbone of the proposed RRC simulation framework by defining how the user equipment (UE) evolves from one RRC state to another over time. At any given instant, the UE exists in exactly one RRC state—Idle, Connected, or Inactive. The transition to the next state is determined not randomly, but through a deterministic function that evaluates the current state and the event occurring in the network. Events such as uplink data arrival, paging reception, or inactivity timer expiration are modelled as discrete inputs that trigger state transitions. This formulation ensures that the UE behaviour remains predictable, repeatable, and compliant with standardized RRC operation [10].
In the context of this project, this equation enables accurate modelling of realistic 5G NR control behaviour within a State flow environment. By enforcing deterministic transitions, the simulation avoids unstable oscillations between states, which are commonly observed in oversimplified models. This is particularly important when evaluating adaptive timer strategies, as the impact of timer tuning must be isolated from unintended state behaviour. The equation also allows the framework to be extended easily to multi-UE or mobility-aware scenarios, making it suitable for large-scale energy efficiency studies.

Inter-Arrival Time Measurement:

$$\Delta t_n = t_n - t_{n-1}$$

This equation calculates the inter-arrival time between consecutive data transmission events initiated by the UE. Inter-arrival time is a fundamental indicator of application behaviour, as it captures how frequently data packets are generated. In practical networks, different services exhibit drastically different inter-arrival characteristics. For example, IoT sensors typically transmit data in short bursts followed by long silence periods, whereas video streaming applications generate continuous traffic with minimal gaps between packets. Accurately measuring this time difference allows the system to infer the nature of the underlying traffic without explicitly classifying the application type [15].
Within the proposed adaptive RRC framework, inter-arrival time acts as the primary observable metric for traffic awareness. Rather than relying on predefined service profiles, the system dynamically learns traffic behavior directly from packet timing. This makes the solution more flexible and applicable to heterogeneous network conditions. By continuously monitoring inter-arrival times, the RRC controller gains the necessary context to decide whether maintaining the Connected state is beneficial or whether transitioning to the Inactive state would yield better energy efficiency.

State flow provides a deterministic execution framework that is well-suited for modelling communication protocols governed by strict transition rules [3]. Each RRC state is represented as a distinct State flow state, and transitions are triggered only by well-defined events such as data arrival, timer expiration, or paging. This approach ensures that the simulated behaviour closely follows the logical structure prescribed by 3GPP specifications [4].

B. RRC State Machine Modeling:

The RRC state machine was modeled to include three core states: Idle, Connected, and Inactive. Each state was designed to reflect the functional and energy characteristics of real 5G NR systems.

In the Idle state, the user equipment (UE) remains disconnected from the network, with minimal power consumption and no retained connection context [7]. Transition from Idle to Connected is triggered by an uplink data request, representing the initiation of communication by the UE.

The Connected state represents active data transmission, where the UE continuously monitors control channels and exchanges user data with the network. Although this state provides low latency and high responsiveness, it also incurs the highest energy consumption. An inactivity timer is initiated once data transmission ceases, governing the transition to the Inactive state [13].

Average Inter-Arrival Time Estimation

$$\bar{\Delta t} = \frac{1}{N}\sum_{i=1}^{N} \Delta t_i$$

This equation computes the average inter-arrival time over a finite observation window, smoothing short-term fluctuations in traffic behavior. Individual packet arrivals can be irregular due to network scheduling, retransmissions, or temporary congestion. Reacting to each arrival independently could lead to unstable RRC behavior, with frequent state toggling. By averaging inter-arrival times, the system obtains a more reliable representation of long-term traffic trends rather than momentary anomalies.

In this project, the averaged inter-arrival time serves as the core input to the Adaptive Inactivity Timer mechanism. It allows the system to distinguish between sustained traffic patterns and transient spikes. This ensures that RRC state transitions are driven by meaningful traffic behavior rather than noise. As a result, the adaptive mechanism achieves better stability while still responding effectively to changes in application activity.

Adaptive Inactivity Timer Formulation:

$$T_{\text{AIT}} = \alpha \cdot \bar{\Delta t} + \beta$$

This equation defines the functional relationship between observed traffic behavior and the inactivity timer duration. The adaptive timer is designed to scale proportionally with average inter-arrival time, ensuring that the RRC controller remains sensitive to traffic dynamics. The scaling factor αdetermines how aggressively the timer responds to changes in traffic, while the constant βintroduces a baseline duration to prevent overly rapid state transitions. Together, these parameters provide a simple yet effective mechanism for timer adaptation.

For this project, the adaptive timer directly addresses the inefficiency of static inactivity timers commonly used in existing networks. When traffic is bursty, the average inter-arrival time increases, causing the timer to shrink and enabling faster transitions to the Inactive state. This significantly reduces radio tail energy consumption. Conversely, when traffic is continuous, the timer expands, minimizing unnecessary state transitions and signaling overhead. This behavior allows the RRC system to balance energy efficiency and responsiveness in a traffic-aware manner.

The Inactive state serves as an intermediate mode in which dedicated radio resources are released while essential connection context is preserved within the radio access network. This state enables faster resumption of communication compared to the Idle state while consuming significantly less energy than the Connected state. Paging events or new data arrivals trigger transitions back to the Connected state [3].

C. Implementation Using State flow:

State flow was used to implement the RRC state machine logic with clearly defined entry, during, and exit actions for each state. Entry actions initialize timers and update energy consumption parameters, while during actions continuously monitor triggering conditions. Exit actions ensure proper cleanup and state variable updates during transitions.

Events such as uplink data arrival, paging, and timer expiration were modeled as discrete inputs to the State flow chart. This event-driven structure allows the simulation to respond immediately to changes in network conditions. Guard conditions were used to ensure that transitions occur only when all specified criteria are satisfied, preventing unintended state oscillations [14].

Adaptive Timer Boundaries:

$$T_{\min} \leq T_{\text{AIT}} \leq T_{\max}$$

This constraint ensures that the adaptive inactivity timer remains within operationally safe limits. Without bounding, extreme traffic patterns could drive the timer to impractical values, resulting in unstable RRC behavior. A very small timer could cause excessive signaling due to frequent transitions, while an excessively large timer could negate energy-saving benefits by keeping the UE connected unnecessarily. Bounding the timer ensures controlled adaptation [8].

In the proposed framework, these bounds are selected based on realistic 5G deployment considerations and 3GPP recommendations. They ensure that adaptive behavior does not violate standard compliance or degrade user experience. This bounded adaptation allows the system to operate reliably across diverse traffic scenarios while maintaining predictable performance.

Composite Network Efficiency Metric:

$$\eta = w_1 \frac{E_{\max} - E}{E_{\max}} + w_2 \frac{L_{\max} - L}{L_{\max}} + w_3 \frac{1}{1 + N}$$

This composite metric provides a unified view of overall network performance by combining energy efficiency, latency, and system stability. Weighting factors allow emphasis on specific performance goals depending on application requirements. By capturing multiple dimensions of performance in a single score, this metric enables fair and meaningful comparison between different RRC timer configurations.

The use of State flow also enables hierarchical and modular modeling, making the system extensible for future enhancements such as mobility management or handover procedures.

D. Adaptive Inactivity Timer Mechanism:

A central component of the proposed methodology is the Adaptive Inactivity Timer (AIT) mechanism. Unlike static timers that use fixed timeout values, the AIT dynamically adjusts the Connected-to-Inactive transition threshold based on observed traffic behaviour.
Traffic profiles were categorized into bursty and continuous patterns. Bursty traffic, commonly associated with IoT and sensor-based applications, is characterized by short data transmissions separated by long idle periods. Continuous traffic, such as video streaming, involves sustained data exchange over extended durations [9].

State-Based Power Consumption Model:

$$P(t) = \begin{cases} P_I, & \text{Idle} \\ P_C, & \text{Connected} \\ P_{IA}, & \text{Inactive} \end{cases}$$

This equation assigns a distinct power consumption level to each RRC state, reflecting the operational characteristics of a 5G UE. In the Idle state, the device performs minimal network monitoring, resulting in low power usage. The Connected state requires continuous control signaling and data transmission, leading to the highest energy consumption. The Inactive state lies between these two extremes, preserving connection context while reducing radio activity.
This power model enables accurate quantification of energy savings achieved through adaptive state management. By explicitly associating power levels with RRC states, the simulation can directly link state residence time to energy consumption. This makes it possible to objectively evaluate how adaptive timers influence battery life, particularly for IoT and mMTC devices where energy efficiency is critical.

The AIT logic monitors inter-arrival times of data packets and adjusts the inactivity timer accordingly. For bursty traffic, the timer duration is reduced to quickly transition the UE into the Inactive state, minimizing radio tail energy consumption [17]. For continuous traffic, the timer is extended to avoid frequent state transitions that could increase signalling overhead and latency. This adaptive behaviour enables a balanced trade-off between energy efficiency and responsiveness [12].

E. Real-Time Energy Consumption Modelling:

To accurately assess the impact of different RRC states and timer strategies, an energy calculation subsystem was integrated directly into the simulation loop. Each RRC state was associated with a predefined power consumption level based on typical 5G UE behaviour [11].

Total Energy Consumption Calculation:

$$E_{\text{total}} = \int_0^T P(t)\, dt$$

This equation computes total energy consumption by integrating power usage over time. Since each RRC state has a different power profile, the total energy depends not only on how often transitions occur but also on how long the UE remains in each state. This formulation captures the cumulative effect of RRC behavior over extended operational periods.
In this project, the energy metric is used as a primary performance indicator to compare static and adaptive timer configurations. By tracking energy consumption continuously, the simulation highlights the direct benefits of faster transitions to the Inactive state under bursty traffic. This approach provides clear evidence of energy savings attributable to adaptive RRC management.

Energy consumption was computed by accumulating power usage over time as the system transitioned between states. This real-time computation allows immediate observation of how state transitions influence overall energy usage, rather than relying on post-simulation estimations. The energy metric is continuously updated and made available for visualization and comparison.

F. Interactive Graphical User Interface:

An interactive graphical user interface (GUI) was developed using MATLAB App Designer to enhance control and observability of the simulation. The GUI functions as a real-time control dashboard, allowing users to manually trigger events such as data requests and paging signals [13].

Discrete-Time Energy Approximation:

$$E = \sum_{k=1}^{K} P(x_k) \cdot \Delta t$$

In discrete-time simulations, energy consumption is calculated incrementally at each simulation step. This equation approximates the continuous energy integral by summing the power consumed during each time interval. This method is computationally efficient and well-suited for real-time simulation environments.
The discrete energy model enables real-time visualization of energy usage within the simulation GUI. As the UE transitions between RRC states, the energy counter updates immediately, allowing users to observe the impact of timer adaptation on power consumption. This real-time feedback enhances both analysis and educational value of the framework.

Visual indicators within the interface display the current RRC state, while time-domain plots illustrate state transitions, timer behaviour, and energy consumption trends. This interactive setup enables intuitive understanding of protocol dynamics and facilitates rapid experimentation with different traffic scenarios and timer configurations.

G. Co-Simulation and Data Exchange

A low-latency, bidirectional communication link was established between the GUI frontend and the Simulink backend. User actions in the GUI are immediately reflected in the State flow model, and simulation outputs are fed back to the interface in real time.

Latency Reduction Metric:

$$\Delta L = L_{\text{setup}} - L_{\text{resume}}$$

This equation quantifies the latency advantage of resuming a connection from the Inactive state compared to performing a full RRC connection setup from Idle. Resume procedures are inherently faster because essential context information is preserved within the network. This reduction in latency is critical for applications that require quick responsiveness.
For this project, the latency reduction metric demonstrates that energy savings do not come at the cost of degraded performance. The adaptive use of the Inactive state enables fast reconnection while still minimizing power usage. This balance is essential for modern 5G services such as IoT control, edge computing, and real-time monitoring.

Signalling Overhead Estimation:

$$O_{\text{sig}} = \sum_i N_i \cdot C_i$$

This equation models the total signaling overhead incurred due to RRC state transitions. Each transition type involves a specific set of control messages, and the cumulative overhead depends on how frequently these transitions occur. Signaling overhead is a critical factor in network scalability and efficiency.
In the proposed adaptive framework, this metric is used to evaluate the trade-off between aggressive energy saving and control signaling cost. While adaptive timers may increase transition frequency under certain traffic conditions, the resulting signaling overhead remains manageable due to reduced full connection setups. This confirms the practicality of the proposed approach [16].

This co-simulation approach ensures synchronized operation between control logic and visualization components. It also allows the system to function as a digital twin of the RRC state machine, providing real-time feedback on protocol behaviour under varying conditions [2].

H. Performance Evaluation Strategy:

The effectiveness of the proposed methodology was evaluated by comparing static inactivity timer configurations with the adaptive timer approach across multiple traffic scenarios. Key performance metrics include energy consumption, latency, signalling overhead, and state transition frequency.

Simulation results were analysed to identify trends and quantify improvements achieved through adaptive state management. This systematic evaluation provides objective evidence of the benefits of the proposed framework and supports its applicability to energy-efficient 5G network design [4].

I. Block Diagram

The 5G NR RRC simulation begins by initializing all required system components and launching the simulation thread. During execution, the system continuously checks for user commands such as start, pause, or stop and processes them accordingly. If the simulation is in the running state, the RRC engine performs periodic tick executions to simulate control plane behavior, and the current state along with execution history is sent to the graphical user interface for visualization. This loop continues until the simulation is terminated, after which final statistics and results are displayed, marking the end of the simulation.

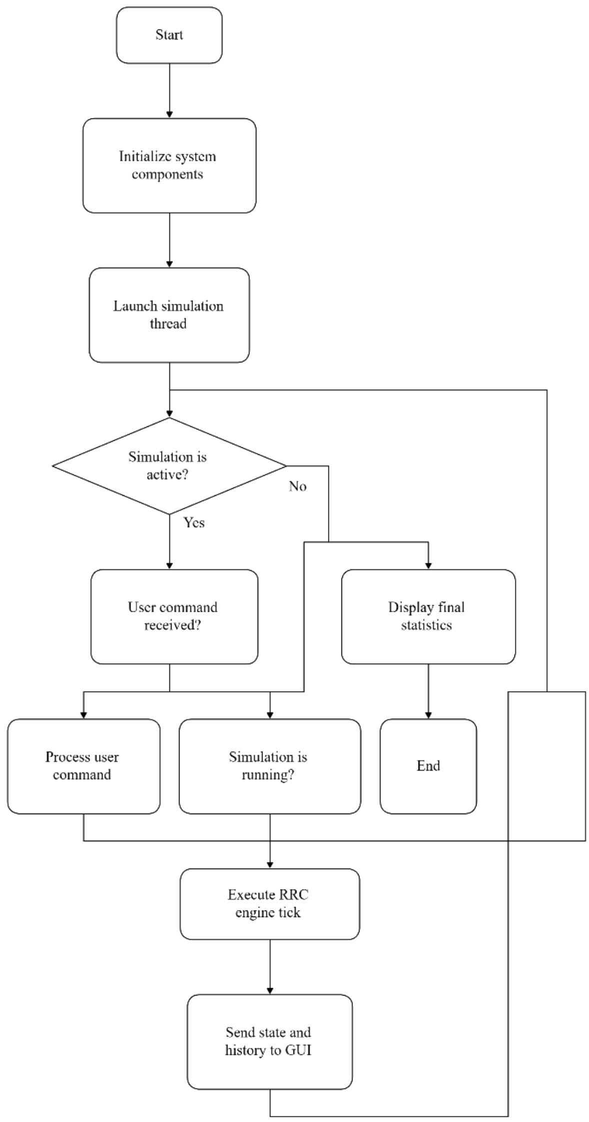

Figure 1: Simulation Flowchart

The simulation begins with a system startup phase, where all essential components required for execution are prepared. During this stage, the simulator configures the core logic responsible for handling RRC state behaviour, initializes the graphical interface, and prepares internal communication mechanisms. These mechanisms allow safe and efficient data exchange between the simulation logic and the user interface, ensuring that the system remains responsive even when complex state transitions occur.
This initialization phase plays a critical role in avoiding inconsistencies later in the simulation. By fully preparing all modules before execution begins, the simulator ensures that user commands, timing logic, and visualization components operate in a synchronized manner. At this point, no simulation activity is performed; instead, the system is placed in a controlled and predictable state, ready to respond to user interaction.

Once initialization is complete, the simulation launches a dedicated execution thread. This thread is responsible for advancing the simulation over time and evaluating the internal RRC logic independently of the graphical interface. Separating the simulation execution from the GUI ensures that real-time visualization does not interfere with timing accuracy or computational consistency [3].

Importantly, the simulation does not begin active execution immediately after the thread is launched. Instead, it enters a paused or inactive mode. This design choice allows users to configure parameters, select traffic profiles, or prepare experimental scenarios without triggering unintended state transitions. The system remains attentive but inactive, waiting for explicit user instruction.
The next stage involves a continuous check to determine whether the simulation is currently active. If the simulation has not been started by the user, the system remains in a listening state. During this time, no RRC processing occurs, and energy consumption or state transitions are not evaluated. This behaviour closely mirrors real network behaviour, where control logic remains dormant until triggered by signalling events or data demand.

If the simulation is marked as active, the execution flow progresses further. This decision point acts as a gatekeeper, ensuring that computational resources are used only when necessary and that simulation behaviour aligns with user intent [2].
One of the key features of the simulator is its ability to process user commands dynamically during execution. When a user command is received such as starting or stopping the simulation, injecting a data burst, simulating a paging event, or modifying traffic u the system immediately processes the request.

Each command influences the internal state machine in a controlled manner. For example, a data burst command may initiate a transition toward a connected state, while a paging event may trigger a wake-up procedure from a low-activity state. By structuring command handling as a distinct phase in the flowchart, the simulator ensures that user interaction is cleanly separated from internal logic execution, reducing complexity and improving clarity [8].

After processing a command, the system evaluates whether the simulation should continue running. If the simulation is paused or stopped, the execution loop does not proceed further and instead returns to monitoring for new user input. This allows the simulator to be halted without losing its internal state or historical data.
If the simulation remains active, the execution advances to the core processing stage, where the actual RRC behaviour is modelled. This conditional check prevents unnecessary computation and ensures that the simulator accurately reflects real-world network behaviour, where state transitions only occur under active conditions [11].

At the heart of the simulator lies the execution of the RRC engine tick. Each tick represents a small, fixed interval of simulated time, during which the system evaluates RRC state logic. During this phase, the simulator determines whether state transitions should occur based on inactivity timers, data availability, and signalling triggers.
In addition to evaluating transitions, the simulator updates internal timers and calculates energy consumption associated with the current RRC state. This allows the framework to model not only functional behaviour but also performance metrics, which are crucial for analysing trade-offs between latency and power efficiency in modern cellular networks.

Following state evaluation, the simulator records the current RRC state and relevant metrics into a historical log. This information is later used to generate visual plots and performance summaries. Periodically, the simulator sends updated state information to the graphical interface, allowing users to observe transitions, activity bursts, and idle periods in real time [5].
The update frequency is carefully controlled to balance responsiveness with computational efficiency. Rather than updating the GUI on every tick, the system refreshes visual elements at predefined intervals, ensuring smooth visualization without overloading the interface.

To ensure realistic behaviour, the simulator enforces a fixed time step between successive ticks. This synchronization mechanism aligns the simulated timeline with real-time execution constraints, making the results easier to interpret and compare across experiments. Maintaining a consistent time base is especially important when analysing delay-sensitive procedures such as paging responses or inactivity-based state transitions.

After completing one execution cycle, the simulator checks once again whether it should continue running. If active, the flow returns to monitoring user commands, allowing interaction and control throughout the simulation lifespan. This looping behaviour enables long-duration experiments and interactive testing without restarting the application [4].

When the simulation is finally terminated, the system exits the execution loop and displays a comprehensive summary of results. These final statistics include overall runtime, cumulative energy consumption, time spent in each RRC state, and the total number of state transitions. Presenting

these results at the end allows users to evaluate system behaviour holistically and draw meaningful conclusions.
The simulation concludes gracefully after displaying the final metrics. All threads are safely terminated, and resources are released. This structured shutdown process ensures data integrity and prepares the system for subsequent simulation runs if needed.

IV. RESULTS AND DISCUSSION

Below figure 1 illustrates the graphical interface and real-time behavior of the proposed 5G NR RRC State Machine Simulator, which was developed to visually demonstrate and analyze RRC state transitions under different traffic conditions. The simulator provides an intuitive representation of how a user equipment (UE) dynamically moves between Idle, Connected, and Inactive states in response to traffic events and network triggers. This visual approach bridges the gap between abstract protocol definitions and practical system behavior, making RRC dynamics easier to interpret and evaluate.

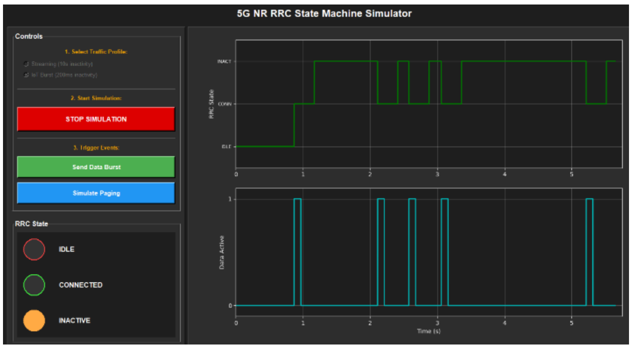

Figure 2: System Demonstration

The left panel of the interface represents the control and interaction section. It allows the user to select predefined traffic profiles, such as continuous streaming traffic or bursty IoT traffic, each associated with different inactivity timer configurations. These profiles emulate real-world application behavior and directly influence the RRC transition logic. The control buttons enable manual triggering of events such as data bursts and paging, allowing controlled experimentation with state transitions. This interactive capability is particularly useful for observing how the system responds to sporadic versus sustained traffic in real time [1].

The bottom-left portion of the interface displays the current RRC state indicator, where each state is represented using a distinct color. Idle is shown in red, Connected in green, and Inactive in orange. This visual feedback provides immediate confirmation of the UE's operational state at any given time. As the simulation progresses, the indicator updates dynamically, reflecting state transitions triggered by traffic activity or inactivity timer expiry. This feature enhances interpretability by making protocol-level decisions visible at a glance.

The upper-right plot presents the RRC state evolution over time. The horizontal axis represents simulation time, while the vertical axis corresponds to the RRC state levels. The plotted trajectory clearly shows transitions from Idle to Connected upon data transmission, followed by transitions to the Inactive state when activity ceases. For bursty traffic scenarios, the UE frequently enters the Inactive state between short data transmissions, demonstrating the effectiveness of the adaptive inactivity timer in reducing unnecessary Connected-state duration. In contrast, during continuous traffic intervals, the UE remains predominantly in the Connected state, indicating stable connectivity and reduced signaling overhead.

The lower-right plot shows the data activity timeline, where active data transmission periods are represented as pulses over time. These pulses correspond directly to user-triggered data bursts or simulated traffic events. By comparing this plot with the RRC state timeline above, the causal relationship between data activity and state transitions becomes evident. Each data pulse triggers a transition to the Connected state, while prolonged inactivity leads to a transition toward the Inactive state. This correlation highlights how traffic behavior directly influences RRC control decisions in the proposed framework [8].

Overall, this figure demonstrates the core contribution of the project: traffic-aware and adaptive RRC state management. The simulator visually confirms that the RRC Inactive state effectively reduces unnecessary energy consumption while preserving fast resume capability. The integration of real-time visualization, user interaction, and protocol-compliant state transitions makes this framework both an analytical tool and an educational platform. The figure validates that adaptive inactivity timer logic aligns RRC behavior with traffic characteristics, thereby achieving an efficient balance between energy efficiency, latency, and signaling stability in 5G NR networks.

The average energy consumption observed under different RRC timer configurations and traffic-aware adaptive strategies. The results clearly show that the Adaptive IoT configuration achieves the lowest energy consumption among all cases, indicating the effectiveness of rapid transitions to the RRC Inactive state for bursty traffic patterns. In contrast, configurations optimized for continuous traffic, such as Adaptive Streaming and Static 10 s timers, exhibit significantly higher energy consumption due to prolonged residence in the Connected state. Static timer configurations demonstrate a gradual reduction in energy usage as the inactivity timer is shortened; however, even the most aggressive static setting fails to match the efficiency of the adaptive IoT strategy. These observations highlight the limitations of fixed inactivity timers and confirm the benefits of traffic-aware adaptive state management [3].

. From the results, it is clearly observed that the Adaptive-IoT configuration consumes the least amount of energy, indicating that intelligent, demand-driven operation significantly reduces unnecessary power usage by activating system resources only when required. In contrast, both the Adaptive-Streaming and Static-10 s configurations exhibit the highest energy consumption, which can be attributed to continuous or rigid operation that keeps the system active for longer durations regardless of actual demand. The Static-1 s configuration shows comparatively lower energy usage than other static approaches because more frequent updates allow the system to react faster to changing conditions, though it still lacks the efficiency of a fully adaptive mechanism. The Static-5 s configuration falls between these extremes, demonstrating that medium update intervals reduce some energy waste but remain less optimal than adaptive strategies. Overall, this output clearly highlights that adaptive control mechanisms, particularly when combined with IoT-based decision-making, play a crucial role in improving energy efficiency compared to fixed, static configurations.

The energy consumption results further emphasize the importance of how system configurations handle resource allocation and operational timing. Configurations that rely on static operation tend to consume more energy because they follow predefined intervals without considering real-time system conditions, which leads to energy being spent even during low-activity or idle periods. This behavior is clearly reflected in the higher energy usage of the Static-10 s and Static-5 s configurations.
On the other hand, adaptive configurations attempt to adjust system behavior dynamically; however, when adaptivity is applied to continuous data-intensive tasks such as streaming, the energy benefits are reduced due to sustained processing and transmission requirements. The Adaptive-IoT configuration stands out because it combines adaptivity with event-driven operation, allowing the system to minimize active time and reduce redundant operations. These observations indicate that energy efficiency is not solely dependent on whether a system is adaptive or static, but also on how intelligently the adaptivity is applied. The output therefore validates the proposed approach of using adaptive, context-aware mechanisms to achieve lower energy consumption while maintaining system performance.

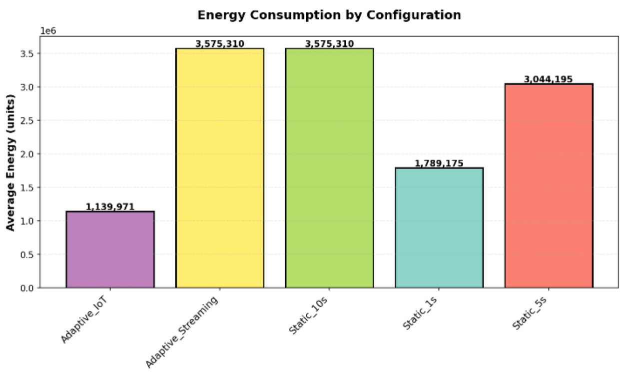

Figure 3: Energy Consumption by Configuration

The percentage distribution of time spent by the UE in Idle, Connected, and Inactive states for each configuration. The Adaptive IoT strategy spends a substantial portion of time in the Inactive state, which directly contributes to reduced energy consumption while still maintaining fast resume capability. Conversely, both Adaptive Streaming and Static 10 s configurations spend over 90% of their operational time in the Connected state, reflecting the requirements of continuous data transmission but at the cost of increased power usage. Static 1 s and Static 5 s configurations exhibit a more balanced distribution, although frequent transitions are observed due to aggressive timer expiration [5]. This state distribution analysis demonstrates how adaptive logic effectively aligns RRC behavior with traffic characteristics, rather than enforcing a uniform policy across all scenarios.

This state distribution output provides deeper insight into how the system behaves over time under different configurations by showing how long it remains in idle, connected, and inactive states. Rather than focusing only on performance metrics, this figure reveals the operational behavior of the system, which directly influences efficiency, responsiveness, and energy usage [15].

In the Adaptive-IoT configuration, the time distribution is more balanced across all three states. A noticeable portion of time is spent in the inactive state, along with a moderate connected duration and a small idle period. This indicates that the system frequently transitions between states based on real-time requirements instead of remaining continuously active. Such behavior reflects an event-driven design where resources are activated only when needed, which explains the lower energy consumption observed earlier. The presence of inactive time suggests that the system intelligently powers down or reduces activity during low-demand periods without compromising functionality.

The Adaptive-Streaming configuration, however, shows a very different pattern. Here, the system spends almost all of its time in the connected state, with negligible idle and inactive durations. This is expected because streaming applications require persistent connectivity and continuous data transfer. While the adaptive mechanism may optimize certain parameters, the nature of streaming workloads forces the system to remain active most of the time. As a result, the state distribution confirms why adaptive streaming does not yield significant energy savings despite being adaptive in nature.

A similar trend is observed in the Static-10 s configuration, where the system also remains predominantly in the connected state. Since the configuration is static, the system does not adapt to changing conditions and continues operating at fixed intervals. This leads to prolonged connectivity even when full activity may not be required. The lack of meaningful inactive time indicates inefficient resource usage, reinforcing the drawbacks of rigid system designs in dynamic environments.

The Static-1 s configuration presents a more mixed behavior. Compared to longer static intervals, the system shows a higher proportion of inactive time along with reduced idle periods. Frequent updates allow the system to respond quickly to state changes, enabling it to disconnect or reduce activity when not required. However, because the configuration is still rule-based rather than adaptive, the transitions are not fully optimized. This results in moderate efficiency improvements, but not to the level achieved by adaptive IoT-based operation [9].

In the Static-5 s configuration, the system exhibits a significant connected duration along with a noticeable inactive portion. This suggests that while the system does enter lower activity states, the fixed update interval causes delayed responses to changing conditions. Consequently, the system remains connected longer than necessary before transitioning, leading to avoidable energy expenditure. The relatively smaller idle time further indicates that the system is frequently active but not always productively so.

Overall, this state distribution analysis highlights that how long a system stays connected or inactive is just as important as how often it updates. Adaptive configurations, especially those driven by IoT-based decision-making, achieve better efficiency by intelligently managing state transitions rather than relying on fixed timing rules. The figure clearly demonstrates that static configurations, regardless of interval length, struggle to balance responsiveness and efficiency. This output therefore reinforces the project's core contribution: intelligent adaptivity leads to more meaningful inactive periods, reduced unnecessary connectivity, and ultimately more efficient system operation.

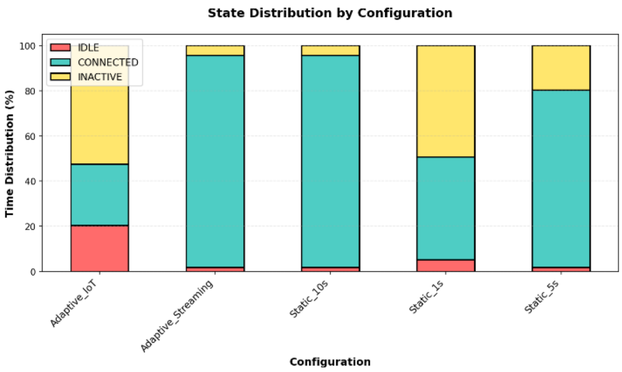

Figure 4: RRC State Distribution by Configuration

The average number of RRC state transitions for different configurations. The Adaptive IoT configuration exhibits the highest transition count, which is expected due to the frequent yet short-lived data bursts typical of IoT traffic. Despite this increase in transitions, the associated energy consumption remains low, indicating that rapid movement into the Inactive state minimizes radio tail energy. Adaptive Streaming and Static 10 s configurations show the lowest number of transitions, as prolonged Connected-state operation avoids frequent state changes. Static 1 s timers result in a noticeable increase in transitions, highlighting the trade-off between aggressive energy saving and signaling overhead [5]. These results emphasize that transition count alone is not a sufficient indicator of inefficiency and must be evaluated alongside energy metrics.

This figure presents the average number of state transitions observed under different system configurations, offering insight into how dynamically each configuration responds to changing conditions. State transitions represent how often the system moves between idle, connected, and inactive states, and therefore reflect the level of adaptability and responsiveness built into each approach.

The Adaptive-IoT configuration exhibits the highest number of state transitions by a significant margin. This indicates that the system frequently adjusts its operating state in response to real-time inputs and environmental changes. Such frequent transitions are characteristic of an intelligent, event-driven system that continuously optimizes its behavior, activating resources when needed and deactivating them when demand drops. While this increases the number of transitions, it enables finer control over resource usage and contributes to improved energy efficiency.

In contrast, both the Adaptive-Streaming and Static-10 s configurations show very few state transitions. This suggests that the system remains in a largely fixed operational state for extended periods, which aligns with continuous data transmission and rigid scheduling, respectively. In these configurations, the lack of frequent transitions indicates limited responsiveness to varying conditions, resulting in prolonged connectivity and reduced opportunities for energy savings [13].

The Static-1 s configuration shows a moderate number of transitions, reflecting more frequent updates compared to longer static intervals. The system is able to react more quickly to changes, leading to more transitions than other static approaches. However, since these transitions are still governed by fixed timing rather than contextual awareness, the adaptability remains limited compared to truly adaptive systems.

The Static-5 s configuration falls between Static-1 s and the more rigid configurations, indicating some degree of responsiveness but with delayed transitions due to the longer update interval. This results in fewer state changes and less efficient adaptation to dynamic conditions.

Overall, this output highlights that a higher number of state transitions is not inherently negative; rather, it signifies intelligent responsiveness when driven by adaptive control. The figure clearly demonstrates that adaptive IoT-based operation prioritizes real-time decision-making, while static and streaming-focused configurations trade adaptability for operational simplicity, often at the cost of efficiency [19].

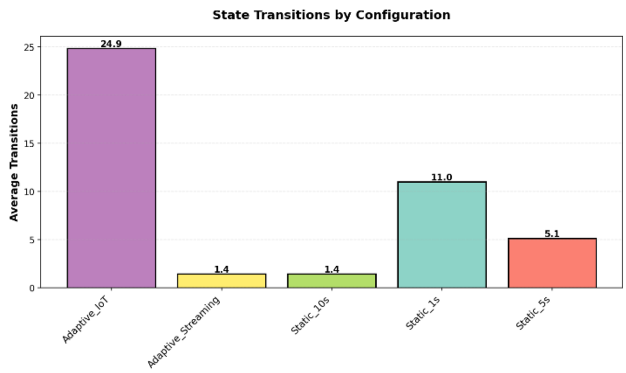

Figure 5: Average RRC State Transitions

The variation in average energy consumption across different traffic categories, including edge computing, IoT, mixed traffic, streaming, and web browsing. Across all categories, adaptive configurations consistently demonstrate lower or comparable energy usage when compared to static timer approaches. The Adaptive IoT strategy shows the most significant savings for IoT and edge traffic, validating its suitability for sporadic transmission patterns. For streaming and web traffic, Adaptive Streaming closely follows the performance of longer static timers, ensuring energy efficiency without compromising session continuity. This figure confirms that adaptive inactivity timers dynamically align system behavior with traffic demand, leading to improved energy efficiency across heterogeneous network conditions [8].

This figure illustrates how average energy consumption varies across different traffic categories for multiple system configurations, offering a clearer understanding of how workload type influences energy behavior. From the trend, it is evident that energy consumption generally increases as the traffic shifts from lighter workloads such as edge and IoT traffic to heavier and more data-intensive categories like fixed, streaming, and web traffic. This pattern highlights the direct relationship between traffic intensity and power usage, as higher data volumes and processing demands naturally require more system resources.

The Static-10 s and Static-5 s configurations consistently show the highest energy consumption across almost all traffic categories. This indicates that static operation with longer fixed intervals leads to inefficient handling of diverse traffic types, as the system remains active for extended periods regardless of actual traffic demand. The sharp rise in energy usage for web traffic in these configurations further emphasizes how static designs struggle to scale efficiently when faced with complex and unpredictable data patterns.

In contrast, the Adaptive-IoT configuration maintains relatively low and stable energy consumption across all traffic categories. Even as the traffic becomes more demanding, the increase in energy usage remains gradual, demonstrating the effectiveness of adaptive, context-aware control. This behavior suggests that the system intelligently allocates resources based on real-time requirements, avoiding unnecessary energy expenditure during low or moderate traffic conditions [10].

The Adaptive-Streaming configuration shows a moderate energy profile, performing better than static configurations but consuming more energy than Adaptive-IoT, particularly for streaming and web traffic. This outcome reflects the continuous connectivity and processing requirements of streaming applications, which limit the extent to which energy savings can be achieved despite adaptivity.

Overall, this output reinforces the idea that both configuration strategy and traffic type play a critical role in determining energy efficiency. Adaptive mechanisms provide clear advantages when handling diverse and dynamic traffic patterns, while static configurations become increasingly inefficient as traffic complexity and intensity increase. The results validate the proposed approach of integrating adaptive control to achieve scalable and energy-efficient system performance across varied traffic scenarios.

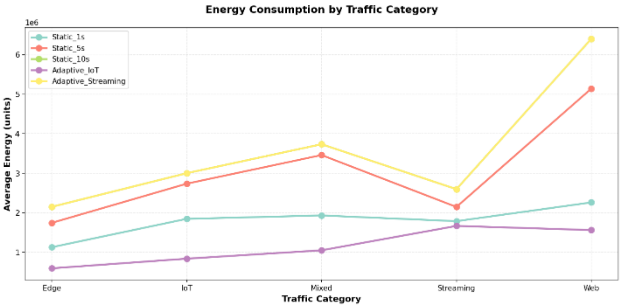

Figure 6: Energy Consumption Across Traffic Categories

The overall network efficiency score for each configuration, combining energy usage, latency, and state stability into a single performance metric. Adaptive Streaming and Static 10 s configurations achieve the highest efficiency scores, indicating optimal performance for sustained data transmission scenarios [1]. The Adaptive IoT configuration scores lower due to its high transition rate; however, this reduction reflects aggressive energy-saving behavior rather than poor system performance. Static 1 s timers fall below the target efficiency threshold, primarily due to excessive signaling overhead caused by frequent state changes. This analysis reinforces the importance of context-aware timer selection rather than relying on uniformly short or long inactivity durations.

This figure illustrates the network efficiency score achieved under different system configurations, providing a consolidated view of how effectively network resources are utilized. The dashed reference line at 50% represents a baseline target efficiency, making it easier to compare how well each configuration performs relative to an acceptable operational threshold. The results show a clear variation in efficiency depending on whether the system follows adaptive or static operational strategies.

The Adaptive-IoT configuration records the lowest network efficiency score, remaining well below the target threshold. This outcome suggests that although the system is highly energy-efficient and adaptive, it prioritizes power savings and selective connectivity over continuous network utilization. As a result, the network resources are not used to their full capacity at all times. This trade-off is expected in IoT-driven systems, where minimizing energy consumption and extending device lifetime often take precedence over maximizing throughput or link utilization [14].

In contrast, both the Adaptive-Streaming and Static-10 s configurations achieve very high network efficiency scores, significantly exceeding the target benchmark. These configurations maintain prolonged connected states and consistent data flow, ensuring that available network resources are almost fully utilized. While this leads to excellent efficiency scores from a network usage perspective, it also explains the higher energy consumption observed earlier, indicating a trade-off between efficiency and power usage.

The Static-1 s configuration lies close to the target efficiency but slightly below it, reflecting a balanced yet imperfect approach. Frequent state updates allow the system to adapt better than longer static intervals, but the absence of intelligent decision-making limits its ability to fully optimize resource usage. This results in moderate efficiency that neither maximizes utilization nor minimizes energy consumption.

The Static-5 s configuration performs comfortably above the target threshold, achieving a strong efficiency score while avoiding the extremes seen in fully static or continuously connected setups. This suggests that medium update intervals can strike a compromise between maintaining network performance and controlling energy usage, though they still lack the fine-grained optimization of adaptive approaches [10].

Overall, this output highlights an important system-level insight: higher network efficiency does not automatically imply better overall performance. Configurations that maximize efficiency often do so at the cost of increased energy consumption, while adaptive IoT-based systems intentionally sacrifice some network utilization to achieve longer operational life and smarter resource management. The figure reinforces the need to evaluate network designs holistically, considering energy efficiency, responsiveness, and utilization together rather than in isolation.

Figure 7: Network Efficiency Score Analysis

A box plot analysis of energy consumption distribution for all configurations, offering insight into variability and worst-case behavior. Adaptive IoT demonstrates the lowest median energy consumption with a narrow interquartile range, indicating both efficiency and stability across simulation runs. Static timer configurations, particularly Static 10 s and Static 5 s, show wider distributions and higher outliers, reflecting increased energy usage during prolonged Connected-state operation. Adaptive Streaming exhibits moderate variability, balancing energy efficiency with consistent performance. This statistical analysis confirms that adaptive timer mechanisms not only reduce average energy consumption but also improve predictability and robustness under dynamic traffic conditions [9].

This box plot analysis presents a detailed view of how energy consumption is distributed for each system configuration, going beyond average values to capture variability, consistency, and the presence of extreme cases. By visualizing the median, interquartile range, and outliers, the figure helps in understanding not just how much energy is consumed, but how stable or unpredictable that consumption is over time.

The static configurations, particularly Static-5 s and Static-10 s, show higher median energy values along with wider spreads. This indicates that these configurations not only consume more energy on average but also experience greater fluctuations in energy usage. The presence of multiple high-value outliers suggests that under certain conditions, such as sudden traffic spikes or inefficient scheduling, static systems can incur significant energy overheads. This variability reflects the inability of fixed-interval operation to adapt to changing network or workload conditions.

The Static-1 s configuration demonstrates a comparatively lower median energy consumption and a tighter interquartile range than other static approaches. More frequent updates allow the system to respond faster to changes, reducing prolonged periods of unnecessary activity. However, some variability still exists, showing that while shorter intervals improve performance, they do not completely eliminate inefficiencies inherent in non-adaptive designs.

The Adaptive-IoT configuration clearly stands out with the lowest median energy consumption and the most compact distribution. The narrow spread indicates highly consistent energy usage, while the limited number of outliers suggests that extreme energy spikes are rare. This behavior highlights the effectiveness of event-driven, adaptive control in maintaining stable and predictable energy consumption, even under varying operational conditions.

The Adaptive-Streaming configuration exhibits a higher median energy level than Adaptive-IoT and a broader spread, reflecting the continuous and data-intensive nature of streaming traffic. Although adaptivity helps in managing resources more efficiently than static approaches, the requirement for sustained connectivity leads to increased energy usage and occasional high-energy outliers [3].

Overall, this box plot reinforces the earlier observations by showing that adaptive configurations not only reduce average energy consumption but also improve stability and predictability. In contrast, static configurations are more prone to energy variability and extreme consumption events, making them less suitable for dynamic and energy-sensitive environments.

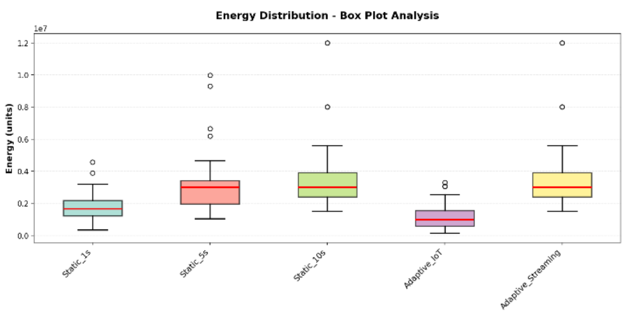

Figure 8: Energy Distribution and Variability Analysis

V. CONCLUSION

This paper presented a comprehensive simulation-based study of Radio Resource Control (RRC) state transitions in 5G New Radio (NR) networks, with a particular focus on evaluating the effectiveness of the RRC Inactive state in improving energy efficiency. By moving beyond the traditional Idle–Connected binary model and explicitly incorporating the Inactive state, the proposed framework provides a more realistic representation of modern 5G network behavior under diverse traffic conditions [19].

A model-based design approach was adopted using MATLAB, Simulink, and Stateflow to ensure deterministic and standards-aligned implementation of the RRC state machine. The simulation accurately captured the dynamic interaction between network events, inactivity timers, and state transitions [16]. The integration of an Adaptive Inactivity Timer (AIT) mechanism enabled the system to dynamically adjust its behavior based on traffic characteristics, addressing the inherent limitations of fixed inactivity timer configurations.

The simulation results clearly demonstrate that adaptive state management leads to substantial improvements in energy efficiency, particularly for bursty and intermittent traffic patterns typical of IoT and mMTC applications. The Adaptive IoT configuration consistently achieved the lowest average energy consumption by rapidly transitioning user equipment into the Inactive state, thereby minimizing radio tail energy. At the same time, the Adaptive Streaming configuration effectively maintained long Connected-state durations for continuous traffic, ensuring low latency and stable performance without unnecessary signaling overhead.

Analysis of RRC state distribution and transition frequency further revealed that aggressive energy saving does not necessarily imply degraded performance. Although adaptive configurations may result in a higher number of state transitions in certain scenarios, these transitions are strategically aligned with traffic demand and do not introduce excessive signaling cost. In contrast, static timer configurations were shown to either waste energy by remaining unnecessarily in the Connected state or incur inefficiencies due to frequent and rigid transitions [19].

The network efficiency score and statistical energy distribution analysis reinforced the advantages of traffic-aware adaptation over one-size-fits-all timer policies. Adaptive configurations demonstrated not only lower average energy consumption but also improved stability and predictability across simulation runs. These findings highlight the importance of intelligent RRC parameter tuning as a key enabler for sustainable and scalable 5G network operation.

Overall, this work validates the RRC Inactive state as a practical and effective mechanism for balancing energy efficiency and responsiveness in 5G NR systems. The proposed simulation framework serves as both a performance evaluation tool and an educational platform for understanding complex protocol behavior [18]. Future work may extend this framework by incorporating mobility management, handover procedures, adaptive DRX cycles, wake-up radio mechanisms, or machine learning-based traffic prediction to further enhance energy efficiency and network intelligence.

The outcomes of this study clearly demonstrate that intelligent RRC state management plays a decisive role in achieving energy-efficient and performance-aware operation in 5G NR networks. By explicitly modeling the RRC Inactive state and evaluating its interaction with different traffic profiles, this work moves closer to representing real-world network behavior rather than relying on simplified assumptions. The results consistently show that static, one-size-fits-all configurations are ill-suited for modern heterogeneous traffic environments, where device behavior can range from sporadic IoT transmissions to continuous high-data-rate streaming [9].

A key takeaway from the analysis is that energy efficiency and network performance are not inherently conflicting objectives. Adaptive configurations are capable of reducing unnecessary energy consumption while maintaining acceptable latency and connectivity, provided that state transitions are aligned with traffic demand. The results highlight that well-designed adaptive mechanisms can intelligently exploit the Inactive state to reduce radio tail energy without causing excessive signaling or instability. In contrast, static configurations either overutilize network resources or fail to respond efficiently to traffic variations, leading to avoidable energy losses.

The comprehensive evaluation using energy consumption trends, state distributions, transition frequencies, network efficiency scores, and statistical variability confirms the robustness of adaptive approaches. These metrics collectively emphasize that adaptive RRC behavior results not only in lower average energy usage but also in more predictable and stable system operation. From a broader perspective, this work underscores the importance of protocol-level intelligence in achieving sustainable 5G deployments, especially as networks continue to support massive numbers of energy-constrained devices [11].

Overall, the proposed simulation framework provides valuable insights into the trade-offs involved in RRC state management and serves as a useful platform for both academic analysis and practical design exploration. The findings reinforce the relevance of the RRC Inactive state as a critical enabler for next-generation energy-aware mobile networks.

- Novelty:

The novelty of this work lies in its holistic and traffic-aware exploration of RRC state management in 5G New Radio networks, with a particular emphasis on moving beyond conventional static and binary state models. While the introduction of the RRC Inactive state has been standardized in 3GPP specifications, its practical implications on energy efficiency, adaptability, and system behavior are often analyzed in isolation or under simplified assumptions. This work distinguishes itself by providing a unified simulation framework that explicitly models Idle, Inactive, and Connected states together, thereby capturing the true dynamics of modern 5G user equipment behavior under heterogeneous traffic conditions.

A key novel aspect of the proposed approach is the integration of an Adaptive Inactivity Timer mechanism that dynamically adjusts state transition behavior based on traffic characteristics rather than relying on fixed timer values. Unlike traditional studies that evaluate static inactivity timers with predefined thresholds, this work demonstrates how traffic-aware adaptation can intelligently balance responsiveness and energy efficiency. By tailoring state transitions to the nature of the traffic—bursty, intermittent, or continuous—the framework enables fine-grained control over radio activity and significantly reduces unnecessary energy consumption without compromising network stability.

Another important contribution of this work is the use of a model-based design methodology to implement the RRC state machine. The combination of MATLAB, Simulink, and State flow allows for a deterministic, modular, and standards-aligned representation of RRC behavior, which is rarely explored in existing literature. This approach not only improves the clarity and correctness of the implementation but also provides a reusable and extensible platform for studying protocol-level optimizations. Unlike black-box simulations or analytical models, the proposed framework allows direct observation of state transitions, timers, and decision logic, offering deeper insight into system behavior.

The novelty is further strengthened by the comprehensive and multi-dimensional evaluation strategy adopted in this study. Instead of relying solely on average energy consumption, the analysis incorporates state distribution, transition frequency, traffic-category-wise energy trends, network efficiency scores, and statistical energy variability using box plot analysis. This layered evaluation reveals subtle trade-offs between energy savings, responsiveness, and network utilization that are often overlooked in single-metric studies. Such an approach enables a more realistic assessment of RRC performance and highlights the limitations of one-size-fits-all configurations.

Additionally, this work uniquely demonstrates that higher adaptability does not necessarily lead to instability or excessive signaling. The results show that frequent state transitions, when driven by intelligent and context-aware policies, can actually improve predictability and efficiency. This challenges the conventional assumption that minimizing transitions is always desirable and instead emphasizes the importance of aligning transitions with traffic demand.

Finally, the proposed framework serves not only as a performance evaluation tool but also as an educational and research-oriented platform. Its modular design allows easy extension to incorporate mobility, DRX mechanisms, wake-up radios, or machine learning-based decision-making. By bridging the gap between theoretical standards and practical behavior, this work provides a foundation for future research in energy-aware RRC optimization and next-generation mobile network design.

- Future Work:

While this study focuses on inactivity timer adaptation and state transitions under controlled traffic scenarios, several promising extensions can further enhance the realism and applicability of the framework. One important direction is the inclusion of user mobility and handover procedures, which would allow investigation of how frequent cell changes affect state transitions, signaling overhead, and energy consumption. This would be particularly relevant for scenarios involving mobile IoT devices and vehicular communications.

Another potential extension involves integrating adaptive Discontinuous Reception (DRX) cycles alongside RRC state management. Coordinating DRX behavior with Inactive state transitions could unlock additional energy savings, especially for devices with periodic but delay-tolerant traffic. Similarly, incorporating wake-up radio mechanisms could further reduce idle listening and improve battery lifetime for ultra-low-power devices.

The framework can also be extended by introducing machine learning-based traffic prediction and decision-making. By learning traffic patterns over time, the network could proactively adjust inactivity timers and state transition policies, enabling more proactive and context-aware optimization. Such intelligence would be particularly beneficial in dense network deployments with highly diverse traffic demands.

Finally, future work may involve validating the simulation results against real-world measurements or testbed implementations to strengthen practical relevance. Extending the model toward 6G-oriented use cases, such as AI-driven services and extreme IoT scalability, would also provide valuable insights into the long-term evolution of RRC state management strategies.

Together, these future directions open pathways for building more intelligent, adaptive, and energy-aware mobile networks that can meet the growing demands of next-generation wireless systems.

VI. References

[1] Y. Liu and H. T. Kung, "Energy Optimization for Mobile Applications by Exploiting the 5G RRC Inactive State," IEEE Transactions on Mobile Computing, vol. 23, no. 4, pp. 2101–2114, 2024.

[2] S. Polaganga and Y. Liang, "Predictive Modeling of RRC Inactive Transitions in 5G Networks," IEEE Communications Letters, vol. 28, no. 2, pp. 356–360, 2024.

[3] J. Zhang, X. Li, and Z. Wang, "Optimizing RRC State Transitions for Machine-Type Communications in 5G,"

IEEE Internet of Things Journal, vol. 10, no. 6, pp. 4982–4994, 2023.

[4] S. Kundu and X. Lin, “Towards Energy-Efficient RAN: From Industry Standards to Trending Practice,” IEEE Communications Magazine, vol. 62, no. 1, pp. 46–52, 2024.

[5] H. Chen and Y. Wu, “Signaling Overhead Analysis of the RRC Inactive State in 5G NR,” IEEE Access, vol. 11, pp. 84210–84222, 2023.

[6] P. Mbulwa, A. R. Ndjiongue, and T. O. Olwal, “Self-Optimization of Handover Control Parameters in 5G Networks,” IEEE Transactions on Network and Service Management, vol. 21, no. 1, pp. 233–245, 2024.

[7] A. Al-Shalabi, M. A. Imran, and Q. Ni, “Adaptive DRX Mechanism for Power Saving in 5G NR RRC Inactive State,” IEEE Transactions on Green Communications and Networking, vol. 7, no. 3, pp. 1456–1468, 2023.

[8] 3GPP, “NR; Radio Resource Control (RRC) Protocol Specification,” 3GPP TS 38.331, Release 17, 2023.

[9] 3GPP, “Wake-Up Signal and Wake-Up Radio for NR,” 3GPP TR 38.858, Release 18, 2024.

[10] R. Gupta and S. Banerjee, “Grant-Free Access for Massive Machine-Type Communications: A Performance Analysis,” IEEE Transactions on Wireless Communications, vol. 21, no. 5, pp. 3274–3286, 2022.

[11] Z. Wang and S. Luo, “RRC Signaling Storm Detection and Mitigation in Open RAN Architectures,” IEEE Network, vol. 37, no. 2, pp. 140–147, 2023.

[12] A. Checko et al., “Cloud RAN for Mobile Networks—A Technology Overview,” IEEE Communications Surveys & Tutorials, vol. 17, no. 1, pp. 405–426, 2015.

[13] S. Sesia, I. Toufik, and M. Baker, 5G NR: The Next Generation Wireless Access Technology, 2nd ed., Academic Press, 2021.

[14] E. Dahlman, S. Parkvall, and J. Sköld, 5G NR: The Next Generation Wireless Access Technology, Academic Press, 2020.

[15] M. Lauridsen, H. Wang, P. Mogensen, and T. B. Sørensen, “Energy-Efficient LTE UE Operation by Adaptive RRC State Transitions,” IEEE Wireless Communications, vol. 21, no. 2, pp. 82–88, 2014.

[16] J. Huang et al., “An Energy-Efficient Small Data Transmission Mechanism for 5G NR,” IEEE Access, vol. 8, pp. 213847–213857, 2020.

[17] X. Lin et al., “5G New Radio: Unveiling the Essentials of the Next Generation Wireless Access Technology,” IEEE Communications Standards Magazine, vol. 3, no. 1, pp. 30–37, 2019.

[18] A. Yadav, R. Budhiraja, and S. Singh, “Traffic-Aware RRC State Management for Energy Optimization in 5G Networks,” Proceedings of IEEE ICC, pp. 1–6, 2022.

[19] M. Simsek et al., “AI-Enabled RAN Optimization for 5G and Beyond,” IEEE Journal on Selected Areas in Communications, vol. 40, no. 1, pp. 286–302, 2022.

[20] MATLAB, “Stateflow and Simulink for Model-Based Design of Communication Systems,” MathWorks White Paper, 2023.