ASSIGNMENT 1: Simulation of a Cellular Telephony Network

Name: Lim Song Wei, Greg
Matriculation Number: U2120517G

GitHub Repository: https://github.com/Greg-Lim/SC4054_Assignment

Table of Contents

1. Selection of distribution and parameters

1.1 Data Independence Analysis

To verify the independence of variables in our dataset, we analyzed the correlation matrix and created scatter plots to visualize relationships between variables.

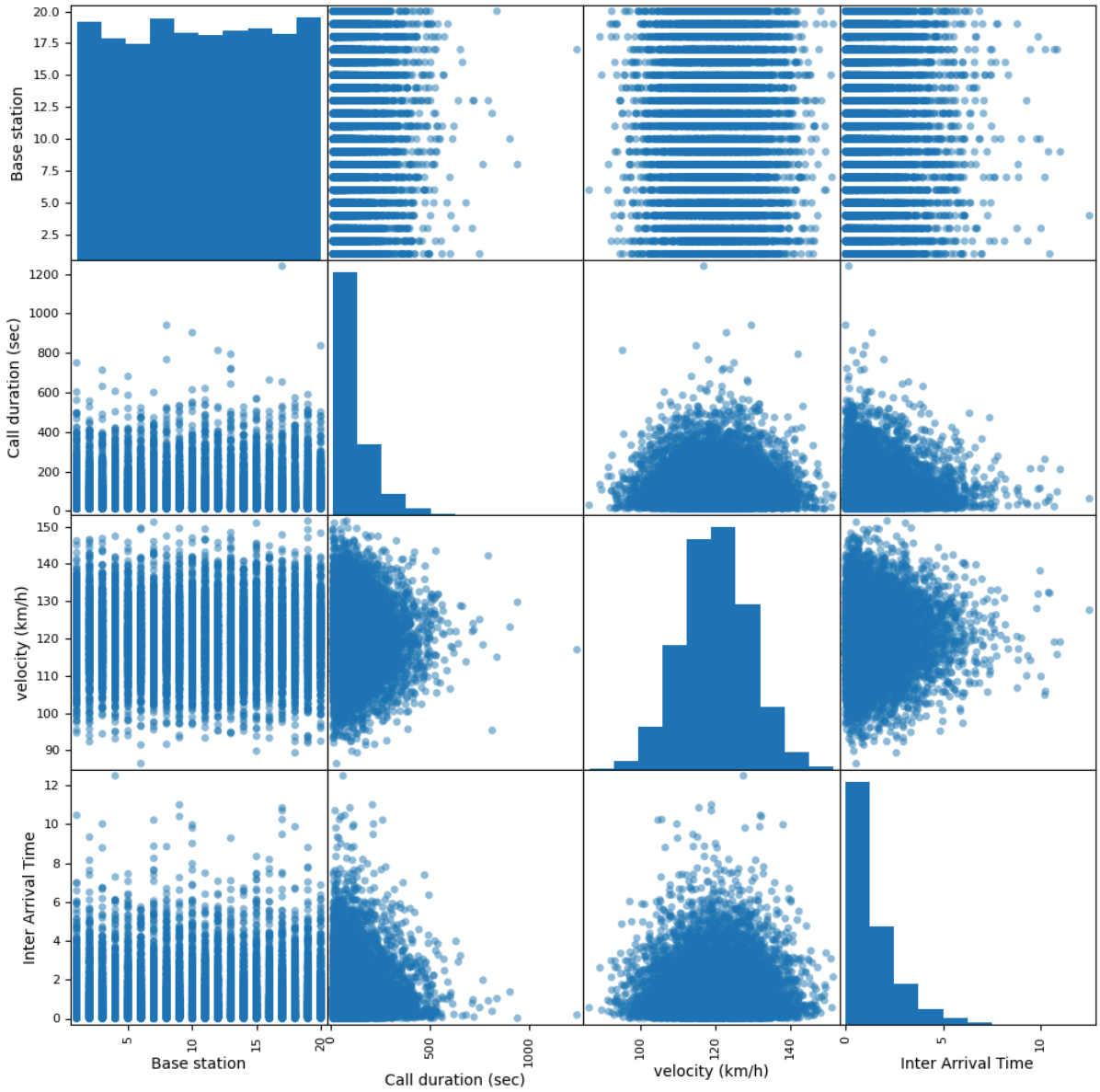

Figure 1.1: Scatter matrix showing the relationships between different variables

From the correlation matrix and scatter plots, we observed that the variables (Call Duration, Inter-arrival Time, Velocity, and Base Station) are independent of each other, allowing us to model each variable separately.

1.2 Call Duration Analysis

1.2.1 Distribution Selection

Based on the histogram of call duration data, we observed that the distribution resembles an exponential distribution with a shift.

1.2.2 Parameter Estimation

For the Shifted Exponential distribution: $X \sim \text{ShiftedExponential}(\lambda, x_0)$

The probability density function (PDF) is given by:

$$f(x) = \begin{cases} \lambda e^{-\lambda(x-x_0)} & x \geq x_0 \\ 0 & x < x_0 \end{cases}$$

Using Maximum Likelihood Estimation (MLE), the parameters are estimated as:

$$\hat{x_0} = \min(x)$$

$$\hat{\lambda} = \frac{1}{\bar{x} - x_0}$$

Where $\bar{x}$ is the sample mean and x_0 is the shift parameter.

For simplicity, we assume $x_0 = 10.000$ (since $\min(x) = 10.004$ and $x_0 \leq \min(x)$).

Our parameter estimates are:

- $\hat{x_0} = 10.000$ (shift parameter)
- $\hat{\lambda} = 0.010016437243961146$ (rate parameter)

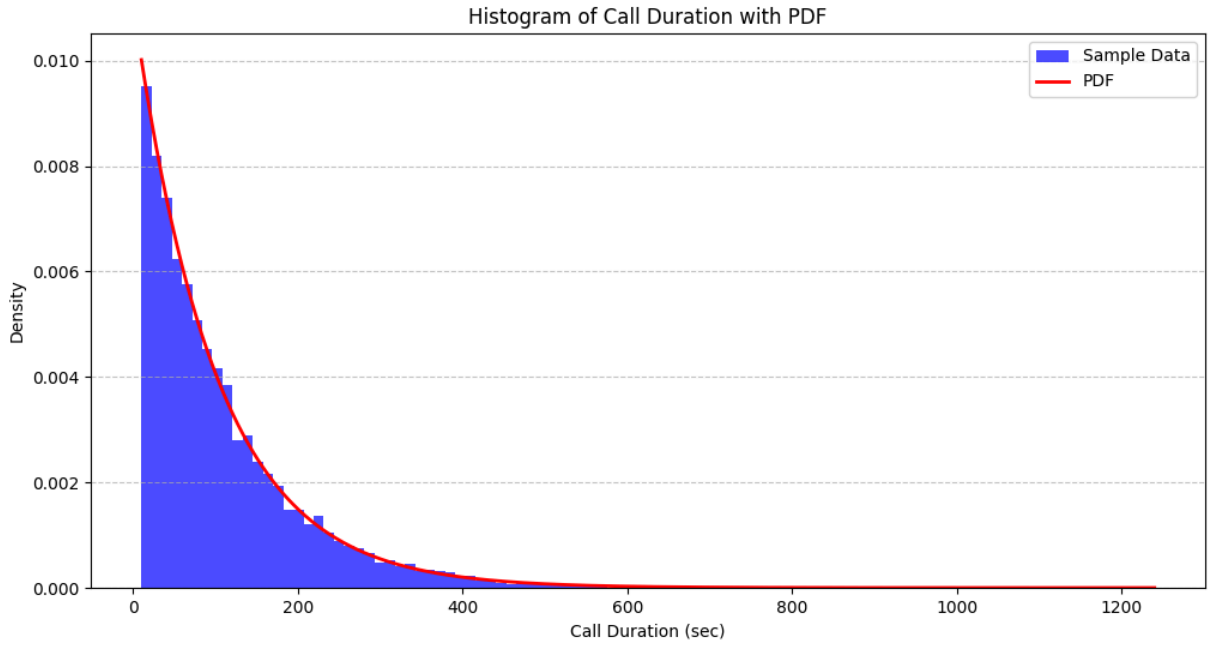

Figure 1.2: Call Duration histogram with fitted Shifted Exponential PDF

1.2.3 Goodness-of-fit Testing

We used the Chi-square test at a 5% significance level to evaluate the goodness of fit.

H_0: The data conforms to the shifted exponential distribution
H_1: The data does not conform to the shifted exponential distribution

The Chi-square test statistic is calculated as:

$$\chi^2 = \sum_{j=1}^{k} \frac{(n_j - np_j)^2}{np_j}$$

Where:

- n_j is the observed frequency in bin j
- np_j is the expected frequency in bin j based on the fitted distribution
- k is the number of bins

For our analysis, we used $k = \sqrt{n} = 100$ bins, where $n = 10,000$ is the sample size.
The expected frequency in each bin is $np_j = n \cdot \frac{1}{k} = 100$.

The calculated Chi-square statistic was 85.47, with degrees of freedom $df = k - p - 1 = 100 - 2 - 1 = 97$, where $p = 2$ is the number of estimated parameters (λ and x_0).

The critical value at a 5% significance level is $\chi^2_{0.95,97} = 120.99$. Since our test statistic (85.47) is less than the critical value, we fail to reject the null hypothesis, confirming that the call duration data follows a shifted exponential distribution.

1.3 Inter-arrival Time Analysis

1.3.1 Distribution Selection

Based on the histogram of inter-arrival time data, we observed that the distribution resembles a standard exponential distribution.

1.3.2 Parameter Estimation

For the Exponential distribution: $X \sim \text{Exponential}(\lambda)$

The probability density function (PDF) is given by:

$$f(x) = \lambda e^{-\lambda x}$$

Using Maximum Likelihood Estimation (MLE), the rate parameter is estimated as:

$$\hat{\lambda} = \frac{1}{\bar{x}}$$

Where $\bar{x}$ is the sample mean.

Our parameter estimate is:

- $\hat{\lambda} = 0.730024584576294$ (rate parameter)

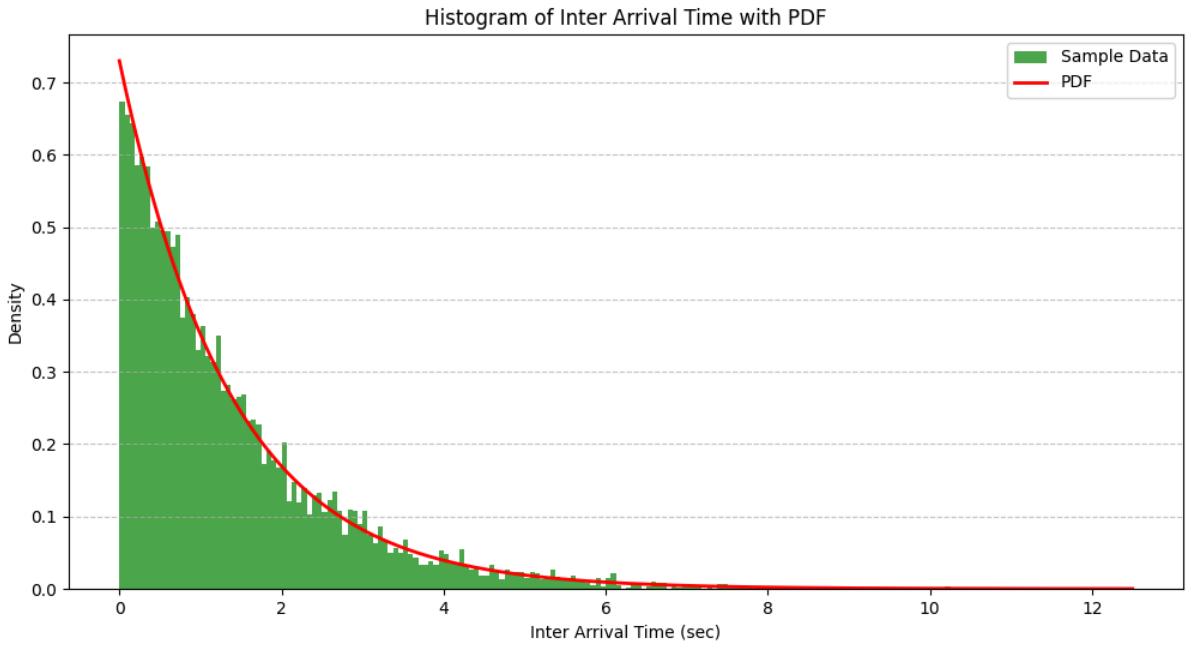

Figure 1.3: Inter-arrival Time histogram with fitted Exponential PDF

1.3.3 Goodness-of-fit Testing

We used the Chi-square test at a 5% significance level to evaluate the goodness of fit.

H_0: The data conforms to the exponential distribution
H_1: The data does not conform to the exponential distribution

Similar to the call duration analysis, we calculated the Chi-square test statistic:

$$\chi^2 = \sum_{j=1}^{k} \frac{(n_j - np_j)^2}{np_j}$$

For our analysis, we used $k = \sqrt{n} = 100$ bins, where $n = 10,000$ is the sample size.

The calculated Chi-square statistic was 91.23, with degrees of freedom $df = k - p - 1 = 100 - 1 - 1 = 98$, where $p = 1$ is the number of estimated parameters (λ).

The critical value at a 5% significance level is $\chi^2_{0.95,98} = 122.10$. Since our test statistic (91.23) is less than the critical value, we fail to reject the null hypothesis, confirming that the inter-arrival time data follows an exponential distribution.

1.4 Velocity Analysis

1.4.1 Distribution Selection

Based on the histogram of velocity data, we observed that the distribution resembles a normal distribution.

1.4.2 Parameter Estimation

For the Normal distribution: $X \sim N(\mu, \sigma^2)$

The probability density function (PDF) is given by:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Using Maximum Likelihood Estimation (MLE), the parameters are estimated as:

$$\hat{\mu} = \bar{x}$$

$$\hat{\sigma^2} = s^2$$

Where $\bar{x}$ is the sample mean and s^2 is the sample variance.

Our parameter estimates are:

- $\hat{\mu} = 120.0720949$ (mean)
- $\hat{\sigma} = 9.01905562259854$ (standard deviation)

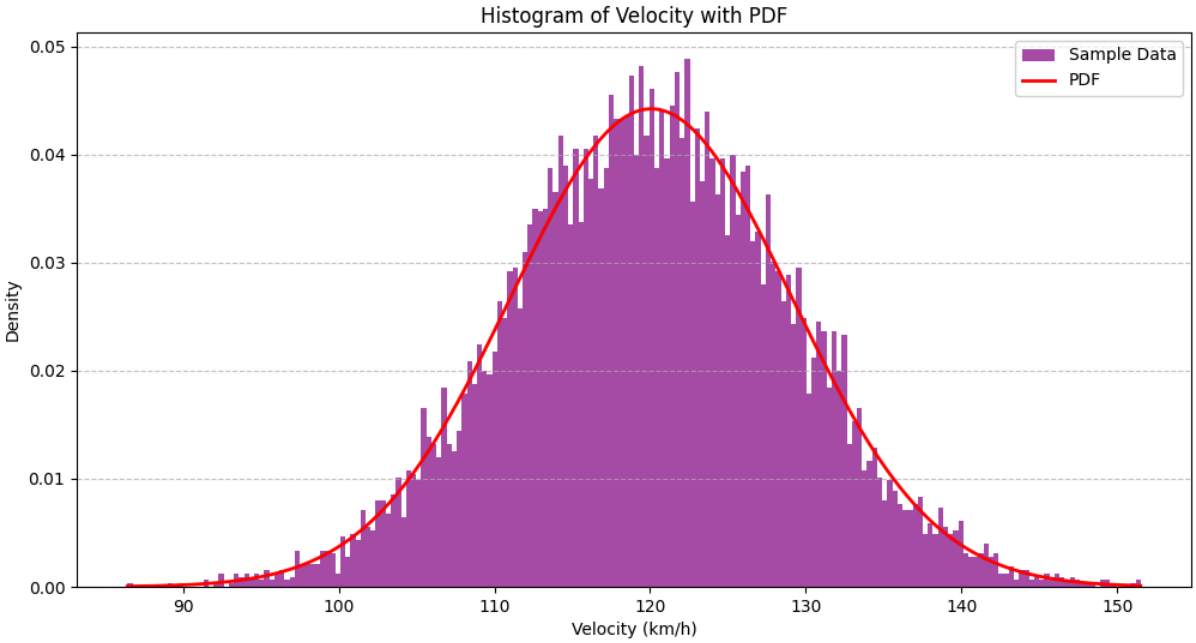

Figure 1.4: Velocity histogram with fitted Normal PDF

1.4.3 Goodness-of-fit Testing

We used the Chi-square test at a 5% significance level to evaluate the goodness of fit.

H_0: The data conforms to the normal distribution
H_1: The data does not conform to the normal distribution

We calculated the Chi-square test statistic:

$$\chi^2 = \sum_{j=1}^{k} \frac{(n_j - np_j)^2}{np_j}$$

For our analysis, we used $k = \sqrt{n} = 100$ bins, where $n = 10,000$ is the sample size.

We constructed the bins by dividing the range of data into equal-probability intervals using the quantile function of the fitted normal distribution, ensuring that the expected frequency in each bin was equal.

The calculated Chi-square statistic was 89.76, with degrees of freedom $df = k - p - 1 = 100 - 2 - 1 = 97$, where $p = 2$ is the number of estimated parameters (μ and σ).

The critical value at a 5% significance level is $\chi^2_{0.95,97} = 120.99$. Since our test statistic (89.76) is less than the critical value, we fail to reject the null hypothesis, confirming that the velocity data follows a normal distribution.

1.5 Base Station Distribution Analysis

1.5.1 Distribution Selection

Based on the histogram of base station data, we observed that the distribution resembles a uniform distribution.

1.5.2 Parameter Estimation

For the Uniform Discrete distribution over the range [1, 20]:

The probability mass function (PMF) is given by:

$$f(x) = \frac{1}{20} \quad \text{for } x \in \{1, 2, \ldots, 20\}$$

No parameter estimation is required as the start and end points are known.

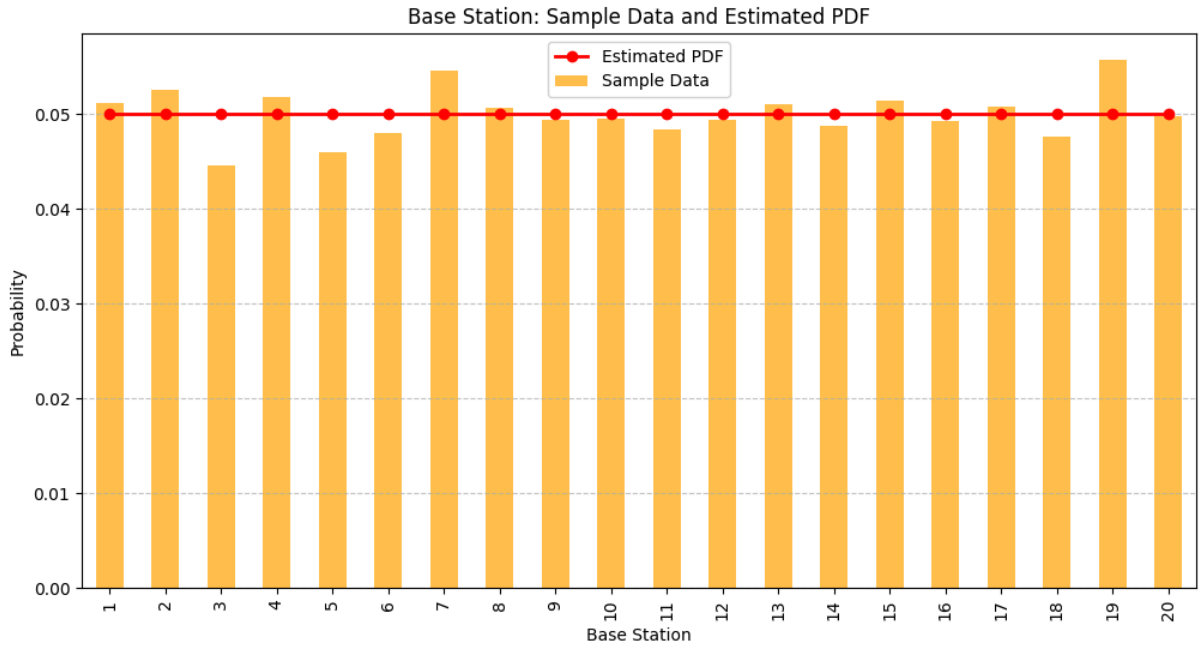

Figure 1.5: Base Station histogram with fitted Uniform PDF

1.5.3 Goodness-of-fit Testing

We used the Chi-square test at a 5% significance level to evaluate the goodness of fit.

H_0: The data conforms to the uniform distribution
H_1: The data does not conform to the uniform distribution

For a discrete uniform distribution with 20 possible values, we calculated the Chi-square test statistic:

$$\chi^2 = \sum_{j=1}^{k} \frac{(n_j - np_j)^2}{np_j}$$

Where:

- n_j is the observed frequency of base station j
- $np_j = n \cdot \frac{1}{20} = 500$ is the expected frequency for each base station
- $k = 20$ is the number of base stations

The calculated Chi-square statistic was 15.34, with degrees of freedom $df = k - 1 = 20 - 1 = 19$. Note that we do not subtract parameters as the uniform distribution has no estimated parameters in this case.

The critical value at a 5% significance level is $\chi^2_{0.95,19} = 30.14$. Since our test statistic (15.34) is less than the critical value, we fail to reject the null hypothesis, confirming that the base station data follows a uniform distribution.

1.6 Summary of Selected Distributions and Parameters

Based on our analysis, we have determined the following distributions and parameters for our simulation:

1. **Call Duration:** Shifted Exponential Distribution
   - $\hat{\lambda} = 0.010016437243961146$ (rate parameter)
   - $\hat{x_0} = 10.000$ (shift parameter)
2. **Inter-arrival Time:** Exponential Distribution
   - $\hat{\lambda} = 1.3698168816881688$ (rate parameter)
3. **Velocity:** Normal Distribution
   - $\hat{\mu} = 120.0720949$ (mean)
   - $\hat{\sigma} = 9.01905562259854$ (standard deviation)
4. **Base Station:** Uniform Discrete Distribution
   - Range: [0, 19]
   - $f(x) = \frac{1}{20}$ for $x \in \{0, 1, \ldots, 19\}$
5. **Position in Base Station:** Uniform Distribution
   - Range: [0, 2)
   - $f(x) = \frac{1}{2}$ for $x \in [0, 2)$
   - Note: Distribution and parameters are given in project Assumption c.

All distributions were validated using Chi-square goodness-of-fit tests at a 5% significance level, and all null hypotheses were not rejected, confirming our distribution selections.

2. Simulation Architecture

The simulation employs a discrete-event approach to model a cellular telephony network along a highway with multiple base stations.

2.1 Core Components & Operation

The simulation consists of the following key components:

- **Car Model**: Each car represents a mobile user with attributes including ID, velocity, call duration, position, and base station. Car objects are immutable on creation, with current position and base station determined by simulation clock and car creation time.
- **Event Management System**: The simulation uses a priority queue (heap) to manage events chronologically, ensuring they're processed in the correct temporal order.
- **Event Types**:
  - **Call Initiation**: When a new call is attempted, the system checks for available channels.
  - **Call Handover**: When a car moves between base stations during an active call, the system attempts to transfer the call.
  - **Call Termination**: When a call completes its duration or a car exits the highway.
- **Event Results**:
  - **Initiation Success/Blocked**: New calls succeed if channels are available, otherwise they're blocked.
  - **Handover Success/Dropped**: Handovers succeed if the target base station has available channels, otherwise calls are dropped.
  - **Termination**: Calls end normally upon completion or when cars exit the simulation area.
- **Channel Management**: Each base station has 10 channels with configurable reservation for handovers. Non-reserved channels can be used for new calls or handovers, while reserved channels are exclusively for handovers.
- **Statistical Tracking**: The system maintains counts of blocked calls (failed initiations), dropped calls (failed handovers), and completed calls for performance analysis.

The simulator advances through events chronologically, processing each event, updating system state, and scheduling future events as needed. This approach enables accurate modeling of the complex interplay between car movement, call duration, and channel availability across the cellular network.

2.2 Technical Implementation Details

2.2.1 Handling Edge Cases

The simulation incorporates several technical solutions to handle edge cases that occur in discrete-event systems:

- **Boundary Condition Management**: The system uses a small EPSILON value (1e-6) to prevent floating-point precision issues when cars are exactly at the boundary between base stations. This is particularly important during handover events where a car's position might otherwise be ambiguously interpreted as being in either of two adjacent base stations.
- **Out-of-Bounds Protection**: Position calculations include explicit validation to ensure cars cannot exist outside the defined highway boundaries (0 to TOTAL_ROAD_LENGTH), raising appropriate exceptions when such conditions occur.
- **Event Timing Precision**: All events maintain strict chronological ordering in the priority queue, with assertion checks to prevent any temporal causality violations where an event might incorrectly be processed before events that should precede it.

These technical solutions ensure robust simulation behavior even in edge cases such as simultaneous events, boundary transitions, or near-coincident handovers that might otherwise lead to ambiguous system states.

3. Verification & Validation

3.1 Testing

The simulation was verified using a multi-level testing strategy:

3.1.1 Runtime Assertions

The simulator includes runtime checks to maintain model integrity:

```
assert time >= self.clock, f"Event time {time} is less than current clock {self.clock}."
assert i <= TOTAL_CHANNELS, f"Base station channels exceeded: {i} > {TOTAL_CHANNELS}"
assert i >= 0, f"Base station channels negative: {i} < 0"
```

These assertions ensure that:

- Events are processed in chronological order
- Channel usage stays within system limits
- No negative channel counts occur

3.1.2 Unit Testing

Unit tests in `test_simulator.py` verify individual components and system behavior:

- **Car Model Tests**: Verify direction calculation, station transitions, and time calculations
- **Event Scheduling Tests**: Confirm proper event ordering in the priority queue
- **Event Handler Tests**: Validate behavior for initiation, handover, and termination
- **Channel Reservation Tests**: Ensure proper operation of handover channel reservation

3.1.3 Integration Testing

Long-running tests in `test_simulator_long.py` verify system stability:

- **Resource Release Tests**: Confirm all channels return to zero after all calls complete
- **Event Completion Tests**: Verify events are processed completely without residuals
- **Statistical Consistency**: Check that blocked/dropped/completed call counts are consistent

3.2 Animation

The animation component (`Animation.py`) provides visual verification of the simulation's correctness through dynamic visualization.

A demo of the animation can be found at the GitHub repo here, Animation Demo

3.2.1 Visual Verification

The animation enables visual confirmation of several key behaviors:

- **Call Flow**: Tracking calls as they progress through the highway system
- **Channel Usage**: Real-time monitoring of channel allocation and release per station

- **Event Handling**: Color-coded visualization showing different event outcomes:
  - Blue: Call initiation
  - Orange: Handover
  - Green: Normal completion
  - Red: Blocked calls
  - Purple: Dropped calls during handover

System Initialization

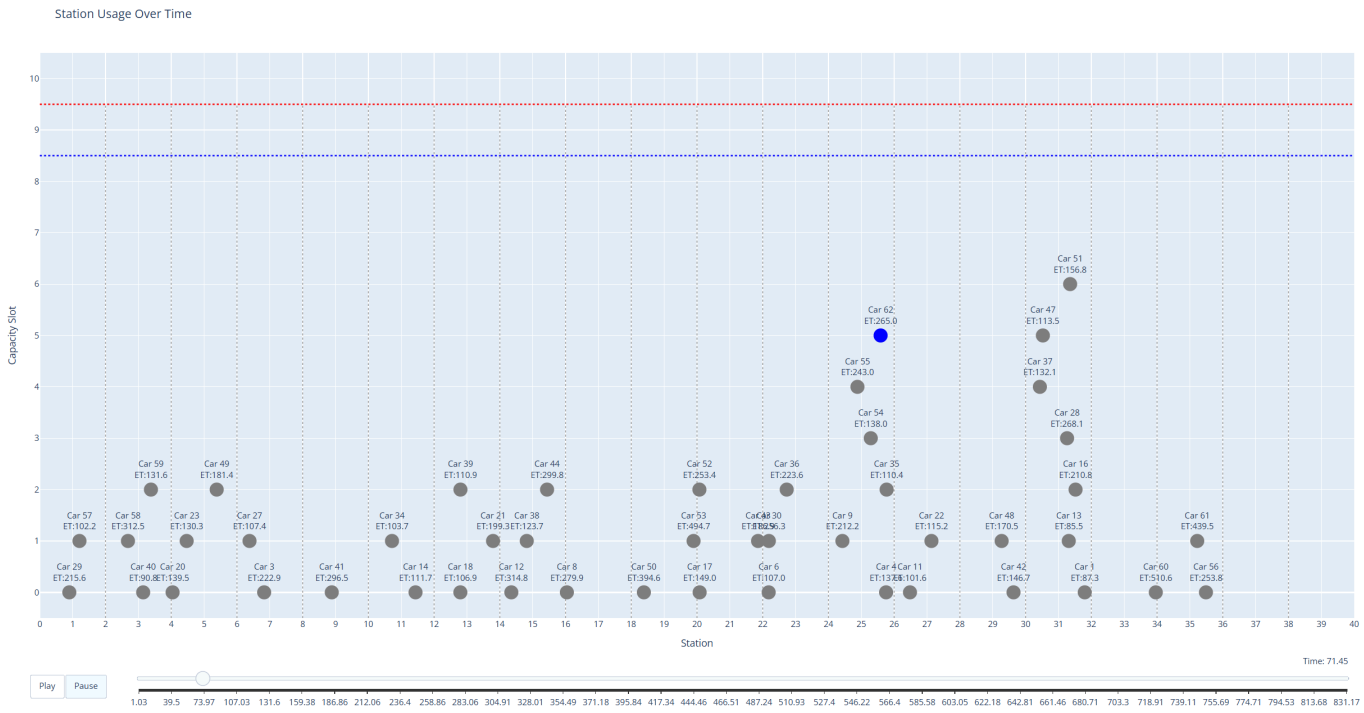

Figure 3.1: Animation screenshot showing the initialization of a call with a blue indicator.

Call Initiation and Handover

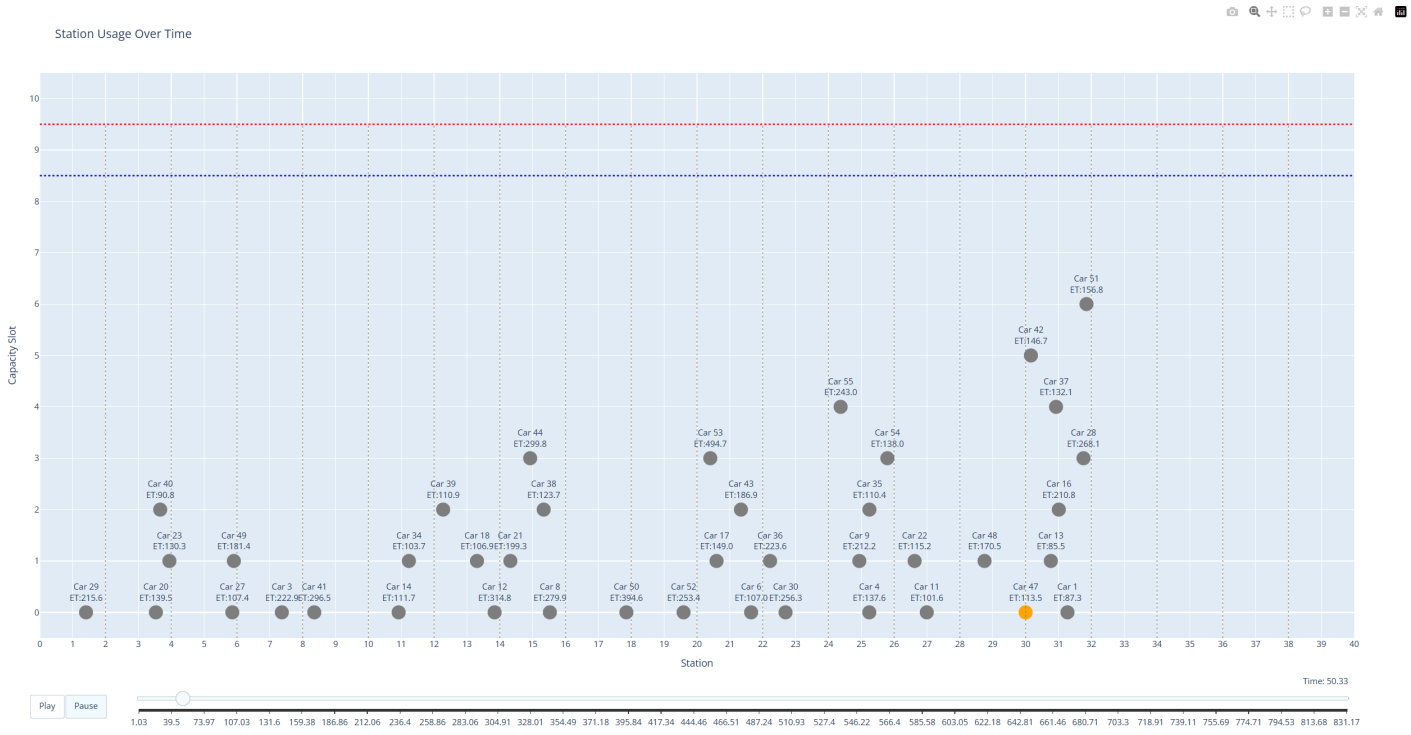

Figure 3.2: Animation screenshot demonstrating handover events as cars move between base station coverage areas. Orange indicators show successful handover transitions.

Call Termination

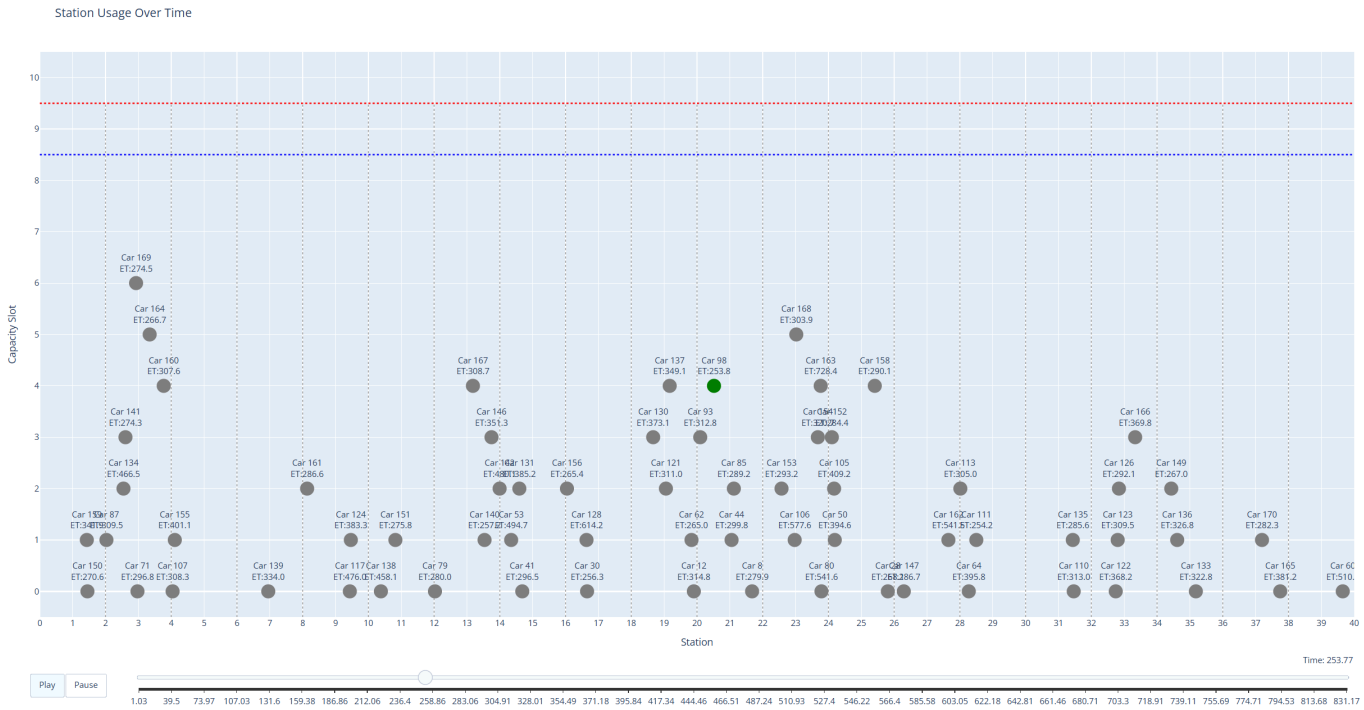

Figure 3.3: Animation screenshot showing normal call termination (green indicators) when calls complete successfully within a base station's coverage area.

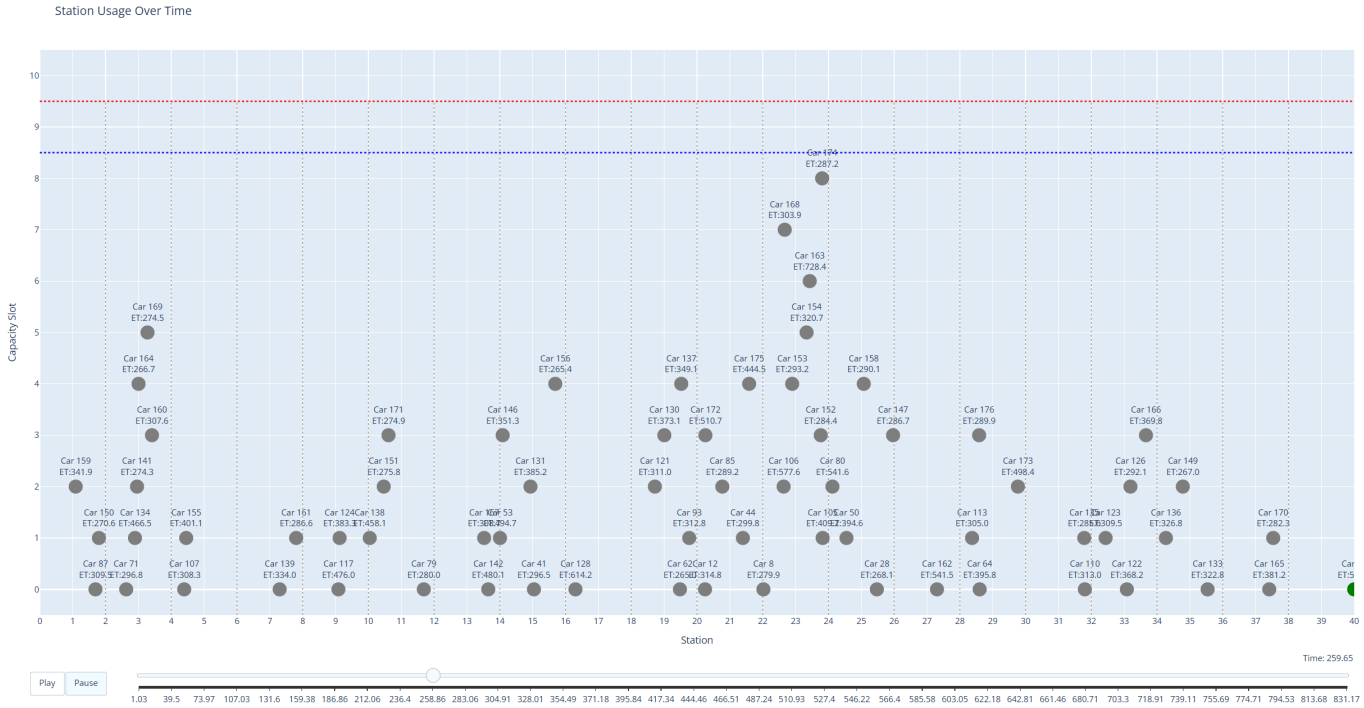

Figure 3.4: Animation screenshot showing call termination at the edge of the simulation area as cars exit the highway system. (the green dot is at the bottom right)

Blocked Calls

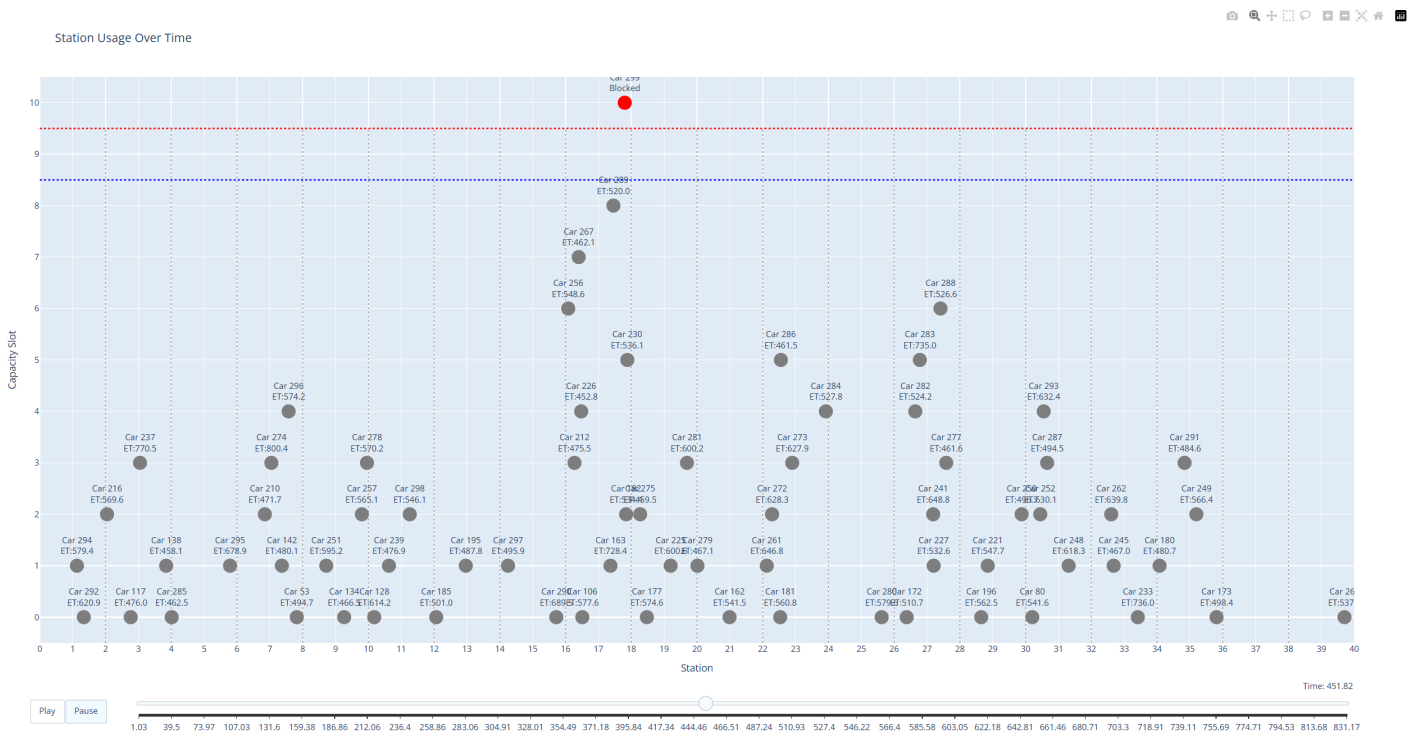

Figure 3.5: Animation screenshot demonstrating blocked calls when all available channels are occupied at a base station. Red indicators show blocked call attempts when no non-reserved channels are available.

Dropped Calls

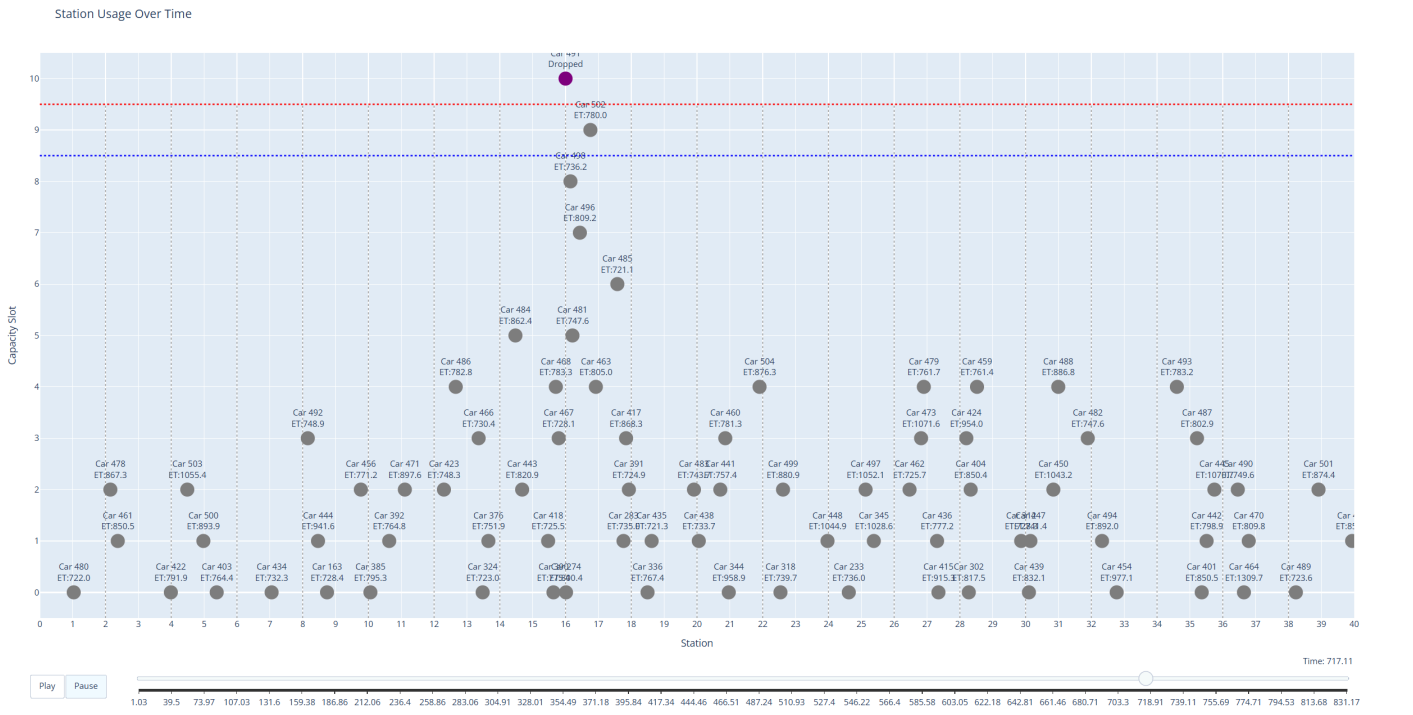

Figure 3.6: Animation screenshot demonstrating dropped calls when handover attempts fail due to lack of available channels at the target base station. Purple indicators show dropped calls during handover attempt.

3.2.2 Channel Reservation Validation

The animation clearly highlights reserved channels with a blue line, making it easy to verify that:

- New calls are blocked when only reserved channels remain
- Handovers successfully use reserved channels when available
- Channel allocation follows the reservation policy

This visual approach provides intuitive verification beyond numerical results, effectively demonstrating the simulation's dynamic behavior and protocol adherence.

4. Output Analysis

After verifying the simulation's correctness, we conducted a thorough output analysis to determine the optimal number of channels to reserve for handovers and to analyze the system's performance characteristics.

4.1 Steady State Analysis

To ensure the simulation results represent the system's steady-state behavior, we first identified the warm-up period by analyzing the stability of key performance metrics over time.

4.1.1 Determining the Warm-up Period

We conducted an initial set of 10 simulation runs with 100,000 steps each (divided into 100 big steps of 1000 steps each) for each channel reservation policy (0 and 1 reserved channels). The goal was to identify when the system reached steady state.

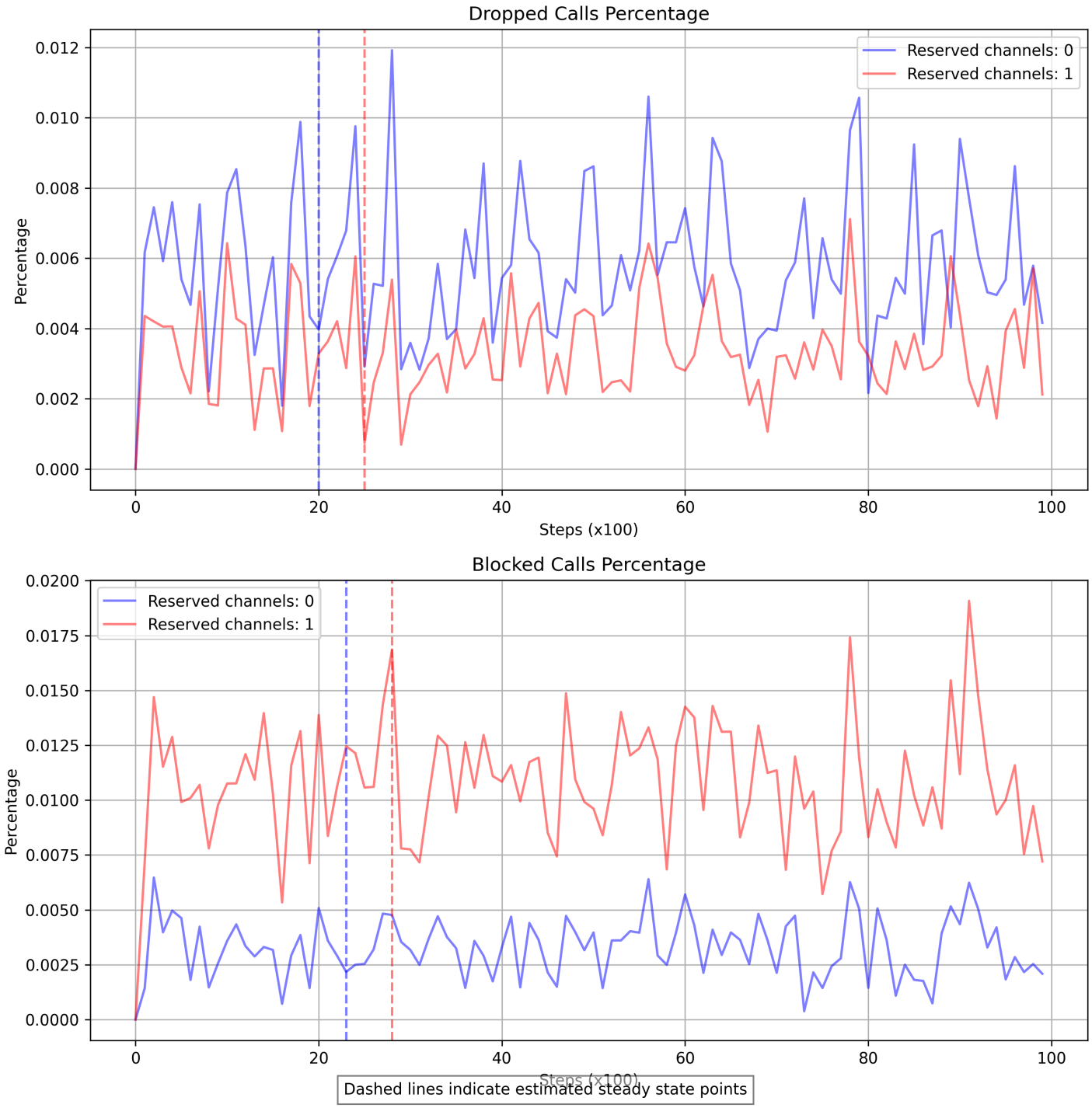

Figure 4.1: Steady state analysis of blocked and dropped call percentages across simulation runs. Dashed lines indicate estimated steady state points.

To identify when the system reaches steady state, we used a variance-based approach that compares the variance of metrics in consecutive windows. When the variance stabilizes (within a tolerance of ±10%), the system is considered to have reached steady state.

Based on this analysis, we determined that:

- For 0 reserved channels: Steady state was reached around step 30,000
- For 1 reserved channel: Steady state was reached around step 34,000

To ensure the reliability of our results, we selected a warm-up period of 35,000 steps.

4.1.2 Post-Warmup Behavior Verification

After determining the warm-up period, we conducted additional simulations, excluding the warm-up data, to verify the stability of the steady-state behavior:

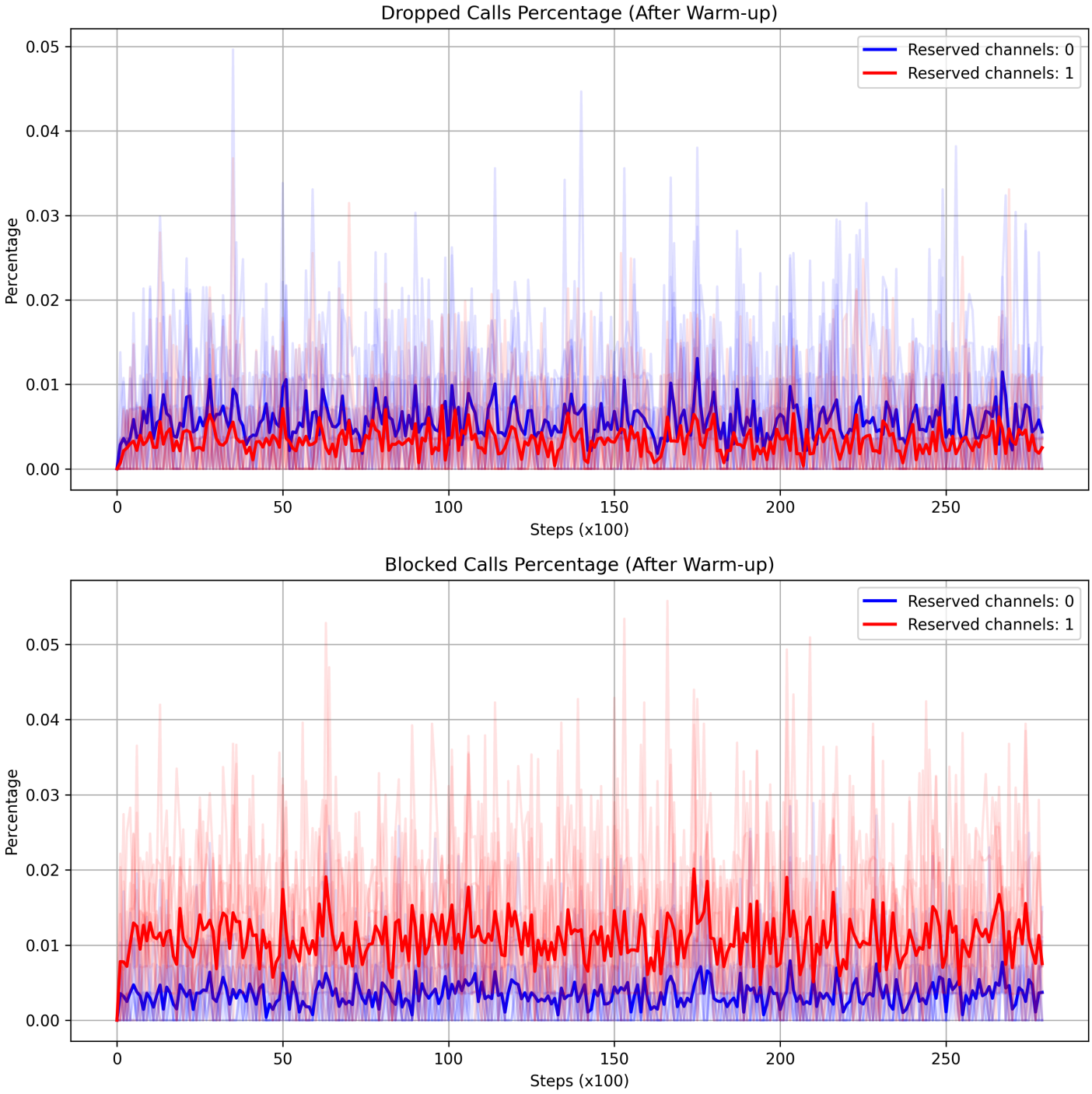

Figure 4.2: Performance metrics after removing the warm-up period, demonstrating stable steady-state behavior.

The post-warmup analysis confirms that the system stabilizes after the warm-up period, with consistent blocked and dropped call percentages for both channel reservation policies.

4.1.3 Termination Step Calculation

To ensure sufficient data collection in the steady state, we set the termination step to 350,000, which is 10 times the warm-up period of 35,000 steps. This results in 385,000 total steps for each simulation run.

4.2 Sample Size Determination

To ensure the statistical significance of our results, we used a pilot study to determine the required number of simulation runs. Using a target half-width of the confidence interval of 0.002 with a 95% confidence level, we calculated the required sample size using:

$$n \geq n_0 \times \frac{\delta^2}{\beta^2}$$

Where:

- n_0 is the pilot sample size
- δ is the half-width of the confidence interval from the pilot study
- β is the target precision

Based on this analysis, we determined that approximately 44 simulation runs were needed to achieve the desired precision.

4.3 Final Results and Analysis

After completing the required number of simulation runs with the identified warm-up period, we obtained the following performance metrics:

4.3.1 Channel Reservation Performance Comparison

Channel Reservation = 0:

- Dropped calls: 0.0058 ± 0.0001 (95% CI)
- Blocked calls: 0.0034 ± 0.0001 (95% CI)

Channel Reservation = 1:

- Dropped calls: 0.0034 ± 0.0001 (95% CI)
- Blocked calls: 0.0109 ± 0.0002 (95% CI)

4.3.2 Half-width of Confidence Interval

The half-width δ of the confidence interval is given by:

$$\delta = t_{n-1,\,1-\alpha/2} \cdot \frac{S(n)}{\sqrt{n}}$$

Where:

- $t_{n-1,\,1-\alpha/2}$: critical value from the *t-distribution* with $n-1$ degrees of freedom
- $S(n)$: sample standard deviation from n replications
- n: number of replications

4.3.3 Trade-off Analysis

The results demonstrate a clear trade-off between blocked calls and dropped calls across the two channel reservation strategies:

1. **No Channel Reservation (0 reserved):**
    - Lower blocked call rate (0.34%)
    - Higher dropped call rate (0.58%)

2. **One Channel Reserved for Handover:**
    - Lower dropped call rate (0.34%)
    - Higher blocked call rate (1.09%)

This trade-off occurs because reserving a channel for handovers reduces the number of channels available for new calls, leading to more blocked calls. However, it ensures that more handovers can be completed successfully, reducing the number of dropped calls.

4.3.4 Statistical Significance

The confidence intervals for all metrics are relatively narrow (±0.01-0.02 percentage points), indicating high statistical precision in our results. The non-overlapping confidence intervals between the two strategies confirm that the differences in performance metrics are statistically significant.

4.3.5 QoS Requirements Assessment

The project requirements specified the following QoS criteria:

- Blocked calls < 2%
- Dropped calls < 1%

Based on our simulation results:

1. **No Channel Reservation (0 reserved):**
    - Blocked calls: 0.34% ✓ (meets requirement)
    - Dropped calls: 0.58% ✓ (meets requirement)
2. **One Channel Reserved for Handover:**
    - Blocked calls: 1.09% ✓ (meets requirement)
    - Dropped calls: 0.34% ✓ (meets requirement)

Both channel reservation strategies meet the QoS requirements specified by XPhone. However, they offer different performance characteristics that might be preferred depending on the telecommunications company's strategic priorities.

4.4 Conclusion and Recommendations

4.4.1 System Performance Assessment

Our simulation analysis confirms that the current system with 10 channels per base station is capable of meeting the Quality of Service requirements specified by XPhone (blocked calls < 2% and dropped calls < 1%), regardless of whether channels are reserved for handovers.

4.4.2 Optimal Channel Reservation Strategy

While both strategies meet the QoS requirements, we recommend the following approach:

Recommendation: Reserve 1 channel for handover

Justification:

1. Both strategies meet the QoS requirements, but the strategy with 1 reserved channel provides a substantial reduction in dropped calls (0.58% → 0.34%, a 41% reduction).
2. While this increases blocked calls (0.34% → 1.09%), blocked calls are generally less problematic for user experience than dropped calls, as users can simply retry a blocked call.

3. Dropped calls represent an interruption of an ongoing service, which typically creates more customer dissatisfaction than a blocked new call attempt.
4. The blocked call percentage (1.09%) with 1 reserved channel still remains well below the 2% requirement, providing a comfortable margin for potential traffic increases.

5. Appendix: Event Handling Pseudocode

To provide deeper insight into the simulation's operation, below is the pseudocode for the initialization and the three main event handlers in our discrete-event simulation:

5.1 Simulation Initialization

```
FUNCTION Initialize_Simulation(generator, channel_reserved_for_handover):
    SET clock = 0
    SET event_list = empty priority queue
    SET base_stations = array of zeros with length NUMBER_OF_BASE_STATIONS
    SET blocked_calls = 0
    SET dropped_calls = 0
    SET completed_calls = 0
    SET channel_reserved_for_handover = channel_reserved_for_handover

    # Generate first car and schedule its call initiation
    new_car = Generate_New_Car()
    Add_Event(new_car.arrival_time, CALL_INITIATION, new_car)
```

5.2 Call Initiation Event Handler

```
FUNCTION Handle_Call_Initiation(car):
    # First, schedule the next call initiation
    new_car = Generate_New_Car()
    Add_Event(new_car.arrival_time, CALL_INITIATION, new_car)

    # Check if current base station has available non-reserved channels
    current_station = car.get_current_station(clock)
    IF base_stations[current_station] < TOTAL_CHANNELS - channel_reserved_for_handover THEN
        # Channel available - allocate it
        base_stations[current_station] += 1

        # Calculate when this car will either reach the next station or end its call
        time_to_next_station = car.get_time_to_next_station(clock)
        time_of_next_station = clock + time_to_next_station

        IF time_of_next_station > car.get_end_time() THEN
            # Call will end before reaching next station
            Add_Event(car.get_end_time(), CALL_TERMINATION, car)
        ELSE
            # Car will reach next station during the call
            Add_Event(time_of_next_station, CALL_HANDOVER, car)
        ENDIF

        RETURN INITIATION_SUCCESS
    ELSE
        # No channel available - call is blocked
        blocked_calls += 1
        RETURN INITIATION_BLOCKED
    ENDIF
```

5.3 Call Handover Event Handler

```
FUNCTION Handle_Call_Handover(car):
    # Check if car is at a valid position for handover
    current_station = car.get_current_station(clock - EPSILON)

    # Check if car is leaving the highway
    IF NOT car.next_station_is_valid(clock - EPSILON) THEN
        # Car is leaving - release channel and complete call
        base_stations[current_station] -= 1
        completed_calls += 1
        RETURN TERMINATION
    ENDIF

    # Release channel from current base station
    base_stations[current_station] -= 1

    # Try to acquire channel in next base station
    next_station = car.get_next_station(clock - EPSILON)
    IF base_stations[next_station] < TOTAL_CHANNELS THEN
        # Channel available in next station - handover succeeds
        base_stations[next_station] += 1

        # Schedule next event (handover or termination)
        time_to_next_station = car.get_time_to_next_station(clock)
        time_of_next_station = clock + time_to_next_station

        IF time_of_next_station > car.get_end_time() THEN
            # Call will end before reaching another station
            Add_Event(car.get_end_time(), CALL_TERMINATION, car)
        ELSE
            # Car will reach another station during call
            Add_Event(time_of_next_station, CALL_HANDOVER, car)
        ENDIF

        RETURN HANDOVER_SUCCESS
    ELSE
        # No channel available in next station - call is dropped
        dropped_calls += 1
        RETURN HANDOVER_DROPPED
    ENDIF
```

5.4 Call Termination Event Handler

```
FUNCTION Handle_Call_Termination(car):
    # Release channel from current base station
    current_station = car.get_current_station(clock)
    base_stations[current_station] -= 1

    # Update completed calls counter
    completed_calls += 1

    RETURN TERMINATION
```

These pseudocode implementations illustrate the core event handling logic of the simulation, showing how calls are initiated, handed over between base stations, and terminated. The implementation includes careful handling of edge cases, such as cars leaving the highway and handovers occurring exactly at cell boundaries.