UNIVERSITY OF PETROLEUM AND ENERGY STUDIES
DEHRADUN

Modeling and Stimulation

Lab Experiment 1

MTECH-COMPUTER SCIENCE ENGINEERING
CYBER SECURITY AND FORENSICS

Name: Jigesh Sheoran
SAP ID: 590025428

Aim: To implement and compare stepped-time simulation and event-based simulation using a traffic signal management system, and analyze their performance, efficiency, and practicality.

Theory :

Stepped-Time Simulation

In stepped-time simulation:

● Time advances in fixed intervals (Δt).

● The system state is checked and updated at every time step, even if no event occurs.

● It is simple to implement but may lead to unnecessary computations.

Example: Checking traffic signal status every second.

Event-Based Simulation

In event-based simulation:

● Time advances only when an event occurs.

● The simulation jumps directly from one event to the next.

● It is more efficient and accurate for systems with discrete state changes.

Example: Traffic signal changes only when its timer expires.

System Description:

The traffic signal system consists of three states:

Signal	Duration (Seconds)
GREEN	30
YELLOW	5
RED	30

The signal cycles continuously in the order:

GREEN → YELLOW → RED → GREEN

Algorithm for Stepped-Time Simulation :

1. Start
2. Initialize simulation time, signal state, and timer
3. Set fixed time step Δt = 1 second
4. While simulation time ≤ total time:
o Display current time and signal state
o Check if state transition condition is satisfied
o Change signal state if required
o Increment simulation time by Δt
5. Stop

Algorithm for Event-Based Simulation :

1. Start
2. Initialize simulation time and initial signal state
3. Schedule first event based on signal timing
4. While simulation time ≤ total time:
o Execute the next scheduled event
o Update signal state
o Schedule next event
o Jump simulation time to event time
5. Stop

Code (Stepped-Time Simulation) :

STS.py > ...

```python
# Author: Jigesh Sheoran
# Last Modified: 02/02/2026

def stepped_time_simulation(total_time, step_size=1):
    print("Stepped-Time Simulation Started\n")

    current_time = 0
    signal = "GREEN"
    time_in_signal = 0

    # durations
    green_time = 30
    yellow_time = 5
    red_time = 30

    checks = 0

    while current_time <= total_time:
        print(f"Time: {current_time}s | Signal: {signal}")

        current_time += step_size
        time_in_signal += step_size
        checks += 1

        # transition logic
        if signal == "GREEN" and time_in_signal >= green_time:
            signal = "YELLOW"
            time_in_signal = 0

        elif signal == "YELLOW" and time_in_signal >= yellow_time:
            signal = "RED"
            time_in_signal = 0

        elif signal == "RED" and time_in_signal >= red_time:
            signal = "GREEN"
            time_in_signal = 0

    print(f"\nTotal State Checks Performed: {checks}")
    print("Stepped-Time Simulation Ended\n")

stepped_time_simulation(total_time=120)
```

Output 1:

```
sheoraninfosec@Jigeshs-MacBook-Air Lab 1 % /usr/local/bin/python3 "/Users/sheoraninfosec
h's MacBook Air/Masters UPES/Semester 2/Modeling and Stimulation/Lab 1/STS.py"
Stepped-Time Simulation Started

Time: 0s | Signal: GREEN
Time: 1s | Signal: GREEN
Time: 2s | Signal: GREEN
Time: 3s | Signal: GREEN
Time: 4s | Signal: GREEN
Time: 5s | Signal: GREEN
Time: 6s | Signal: GREEN
Time: 7s | Signal: GREEN
Time: 8s | Signal: GREEN
Time: 9s | Signal: GREEN
Time: 10s | Signal: GREEN
Time: 11s | Signal: GREEN
Time: 12s | Signal: GREEN
Time: 13s | Signal: GREEN
Time: 14s | Signal: GREEN
Time: 15s | Signal: GREEN
Time: 16s | Signal: GREEN
Time: 17s | Signal: GREEN
Time: 18s | Signal: GREEN
Time: 19s | Signal: GREEN
Time: 20s | Signal: GREEN
Time: 21s | Signal: GREEN
Time: 22s | Signal: GREEN
Time: 23s | Signal: GREEN
Time: 24s | Signal: GREEN
Time: 25s | Signal: GREEN
Time: 26s | Signal: GREEN
Time: 27s | Signal: GREEN
Time: 28s | Signal: GREEN
Time: 29s | Signal: GREEN
Time: 30s | Signal: YELLOW
Time: 31s | Signal: YELLOW
Time: 32s | Signal: YELLOW
Time: 33s | Signal: YELLOW
Time: 34s | Signal: YELLOW
Time: 94s | Signal: GREEN
Time: 95s | Signal: YELLOW
Time: 96s | Signal: YELLOW
Time: 97s | Signal: YELLOW
Time: 98s | Signal: YELLOW
Time: 99s | Signal: YELLOW
Time: 100s | Signal: RED
Time: 101s | Signal: RED
Time: 102s | Signal: RED
Time: 103s | Signal: RED
Time: 104s | Signal: RED
Time: 105s | Signal: RED
Time: 106s | Signal: RED
Time: 107s | Signal: RED
Time: 108s | Signal: RED
Time: 109s | Signal: RED
Time: 110s | Signal: RED
Time: 111s | Signal: RED
Time: 112s | Signal: RED
Time: 113s | Signal: RED
Time: 114s | Signal: RED
Time: 115s | Signal: RED
Time: 116s | Signal: RED
Time: 117s | Signal: RED
Time: 118s | Signal: RED
Time: 119s | Signal: RED
Time: 120s | Signal: RED

Total State Checks Performed: 121
Stepped-Time Simulation Ended
```

Code (Event-Based Simulation) :

EBS.py > ...

```python
# Author: Jigesh Sheoran
# Last Modified: 02/02/2026

def event_based_simulation(total_time):
    print("Event-Based Simulation Started\n")

    current_time = 0
    signal = "GREEN"

    # durations
    durations = {
        "GREEN": 30,
        "YELLOW": 5,
        "RED": 30
    }

    # transition order
    next_signal = {
        "GREEN": "YELLOW",
        "YELLOW": "RED",
        "RED": "GREEN"
    }

    events = 0

    while current_time <= total_time:
        print(f"Time: {current_time}s | Signal: {signal}")

        # jump to next event
        current_time += durations[signal]
        signal = next_signal[signal]
        events += 1

    print(f"\nTotal Events Processed: {events}")
    print("Event-Based Simulation Ended\n")

event_based_simulation(total_time=120)
```

Output 2:

```
sheoraninfosec@Jigeshs-MacBook-Air Lab 1 % /usr/local/bin/python3 "/Users/sheoraninfosec/D
h's MacBook Air/Masters UPES/Semester 2/Modeling and Stimulation/Lab 1/EBS.py"
Event-Based Simulation Started

Time: 0s | Signal: GREEN
Time: 30s | Signal: YELLOW
Time: 35s | Signal: RED
Time: 65s | Signal: GREEN
Time: 95s | Signal: YELLOW
Time: 100s | Signal: RED

Total Events Processed: 6
Event-Based Simulation Ended
```

Observation Table:

Parameter	Stepped-Time Simulation	Event-Based Simulation
Time advancement	Fixed (1 second)	Event-driven
Signal state checks	Every second	Only on events
Total simulation time	120 seconds	120 seconds
Total state checks / events	121 checks	6 events
Signal transitions	Detected with step delay	Exact transition time
Redundant operations	High	None
CPU/log output	Large	Minimal

Signal Transition Observations:

Signal	Stepped-Time Output	Event-Based Output
GREEN	Printed every second (0–29s)	Printed once at 0s
YELLOW	Printed every second (30–34s)	Printed once at 30s
RED	Printed every second (35–64s, etc.)	Printed once at 35s
Cycle repetition	Continuous step-wise	Jump-based

Results

1. The stepped-time simulation performed 121 state checks for a total simulation duration of 120 seconds.
2. The event-based simulation processed only 6 events, corresponding to actual signal transitions.
3. Both simulations produced logically correct signal sequencing.
4. The stepped-time simulation resulted in significant redundant computations.
5. The event-based simulation achieved the same functional behavior with significantly fewer operations.

Conclusion

Both stepped-time and event-based simulations were implemented successfully for the traffic signal system. Stepped-time simulation is simple and easy to understand but performs frequent, unnecessary state checks, resulting in higher computational overhead. Event-based simulation processes only actual signal transitions, making it more efficient and accurate.

Hence, for systems with discrete state changes like traffic signals, event-based simulation is more practical and scalable, while stepped-time simulation is better suited for learning and small-scale applications.

—----------END OF DOCUMENT —----------