



CZ4015 Simulation & Modelling

Assignment 2

**Submitted by:** Lim Zhi Qing

**Matriculation Number:** U2021897L

**NANYANG TECHNOLOGICAL UNIVERSITY**  
**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

## System State Variables

```
1 Integer: Directions      /* number of directions */
2 Integer: Cells           /* number of cells */
3 Integer: CellSize        /* diameter of each cell */
4 Integer: Channels        /* number of channels per cell */
5 Integer List: CellCalls  /* number of active calls for each cell */

6 Integer: TotalCalls      /* total number of calls */
7 Integer: DroppedCalls    /* number of dropped calls */
8 Integer: BlockedCalls    /* number of blocked calls */

9 LinkedList: FEL
```

## Main Process

```
/* Initialize state variables*/
1 Directions = 2;
2 Cells = 20;
3 CellSize = 2000;
4 Channels = 10;
5 CellCalls = List[Cells];

6 TotalCalls = 0;
7 DroppedCalls = 0;
8 BlockedCalls = 0;

9 While (terminating condition is not true) {
10     Hold(A); /* inter-arrival time */
11     Create and activate a call process;
12 }

/* Compute simulation statistics */
```

## Call Process

```
1 Integer: Cell = Distribution(X);
2 Integer: Direction = UniformDistribution(Directions);
3 Integer: Position = UniformDistribution(Cells);
4 Integer: Speed = Distribution(Y);
5 Integer: Duration = Distribution(Z);
6 Integer: TravelTime;

7 TotalCalls = TotalCalls + 1;

8 If (CellCalls(Cell) >= Channels) {
9     BlockedCalls = BlockedCalls + 1; /* block call */
10    Terminate Call Process;
11 }
12 Else {
13     CellCalls(Cell) = CellCalls(Cell) + 1; /* accept call */
14     TravelTime = Position / Speed;
15     If (Duration < TravelTime) {
16         TravelTime = Duration;
17     }
18 }

19 Hold(TravelTime);
20 Duration = Duration - TravelTime;

21 While (Duration > 0) {
22     CellCalls(Cell) = CellCalls(Cell) - 1; /* deallocate call */
23     Cell = Cell + Direction; /* get next cell */
24     If (CellCalls(Cell) >= Channels) {
25         DroppedCalls = DroppedCalls + 1; /* drop call */
26         Terminate Call Process;
27     }
```

```

28     Else {
29         CellCalls(Cell) = CellCalls(Cell) + 1; /* handover call */
30         TravelTime = CellSize / Speed;
31         If (Duration < TravelTime) {
32             TravelTime = Duration;
33         }
34         Hold(TravelTime);
35     }
36     Duration = Duration - TravelTime;
37 }

38 Terminate Call Process;

```

## Hold

```

Hold(T)
{
    Put Resume-event with time Now+T into FEL;
    Suspend execution of thread;
    Transfer to scheduler;
}

```

## Scheduler

```

Scheduler
{
    If (sim not done) {
        Remove event from FEL;
        Call event handler;
    }
}

```

## Resume-Event Handler

```
Resume-Event Handler
{
    Activate process from Hold/Wait point;
}
```