5.1 M/M/1 Queuing model with dynamic adjustment.

Algorithm Steps:

Algorithm consists of 5 stages. Stage 1 includes calculating utilization, average waiting time and task parameters.

Second stage includes Checking Data complexity, if data is found to be complex, then offload it to the cloud, else step forward for local processing.

Third includes calculating Weighted score by considering the task parameters of the process that includes:

Cost : Higher cost is more preferable, , so positive weight. (w1 x Cost)

Communication Band: Higher bandwidth is generally preferable, , so positive weight (+w2 x Communication Band)

Baud Rate: Higher baud rate is generally preferable, so positive weight. (+w3 × Baud Rate)

Signal Level: Higher signal level is generally preferable, so positive weight (+w4 × Signal Level)

Latency: Lower latency is generally preferable, so negative weight (-w5 × Latency)

If Calculated score is Greater than Threshold Score, then proceed for local processing else offload to cloud.

Stage 4 includes Finding Queue Length, if queue length is less than Threshold Queue length && Average waiting time < Threshold waiting time, then this task can be added to the queue for local processing, else offload it to cloud.

Then final stage consists of Dynamic adjustment which includes recalculating system Parameters i.e. Utilization & Average Waiting Time.

ALGORITHMS

MAIN ALGORITHM FOR CLOUD FOG

```
OFFLOADING procedure
CLOUD_FOG_OFFLOADING_ALGO(CFOA)
        Calculate Utilization ➡️ ρ
        Average Waiting Time ➡️ Wq
        Calculate Task Personalized Parameters
                Task_Param = [CT, CmBW, BR, SGL, L]
        Check Data Complexity
                if Data is complex then Offload to cloud
                else Calculate Weighted_Score (Task_Param)
                        if Weighted Score > Threshold Score then
                                Check Queue Length and update queue
                                for each task in Queue
                                        if task is completed then
                                                Remove it from queue
                                                Update
                                                Queue_Length
                                                If Queue_Length>0 then
                                                Queue_Length--
                                        if Queue_Length < Threshold_Queue_Length && Average
                                        waiting time < Threshold waiting time then
                                                Add the Process to the queue.
                                                Update Queue_Length
                                                        Queue_Length++
                                                Process Locally on Fog
                                        else Offload to cloud
                                end for
                        else Offload to Cloud
        Recalculate Utilization
        Recalculate Waiting Time
        Back to Procedure
end procedure
```

SUBROUTINE FUNCTIONS

procedure WEIGHTED_SCORE(array Task_Param)
    Assign the weight to each parameters (w1, w2, w3, w4, w5)
    Calculate Weighted Score
    WSc = w1×CT + w2×CmBW + w3×BR + w4×SGL - w5×L
    RETURN WSc
end procedure

procedure DATA_COMPLEXITY
    Check Data type ➡️ dTyp
        If dTyp includes [videos, bulk images, audio, bulk unstructured data]
            RETURN Complex
        else RETURN Not Complex
end procedure