The Fifth International Conference on Fog and Mobile Edge Computing (FMEC 2020)
June 30th - July 3rd, 2020
Paris, France

A random walk based load balancing algorithm for Fog Computing

Roberto Beraldi*, Claudia Canali˟, Riccardo Lancellotti˟, Gabriele Proietti Mattia*

*Department of Computer, Control and Management Engineering "Antonio Ruberti", Sapienza University of Rome, Italy
˟Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, Modena, Italy

proiettimattia@diag.uniroma1.it

Outline

1

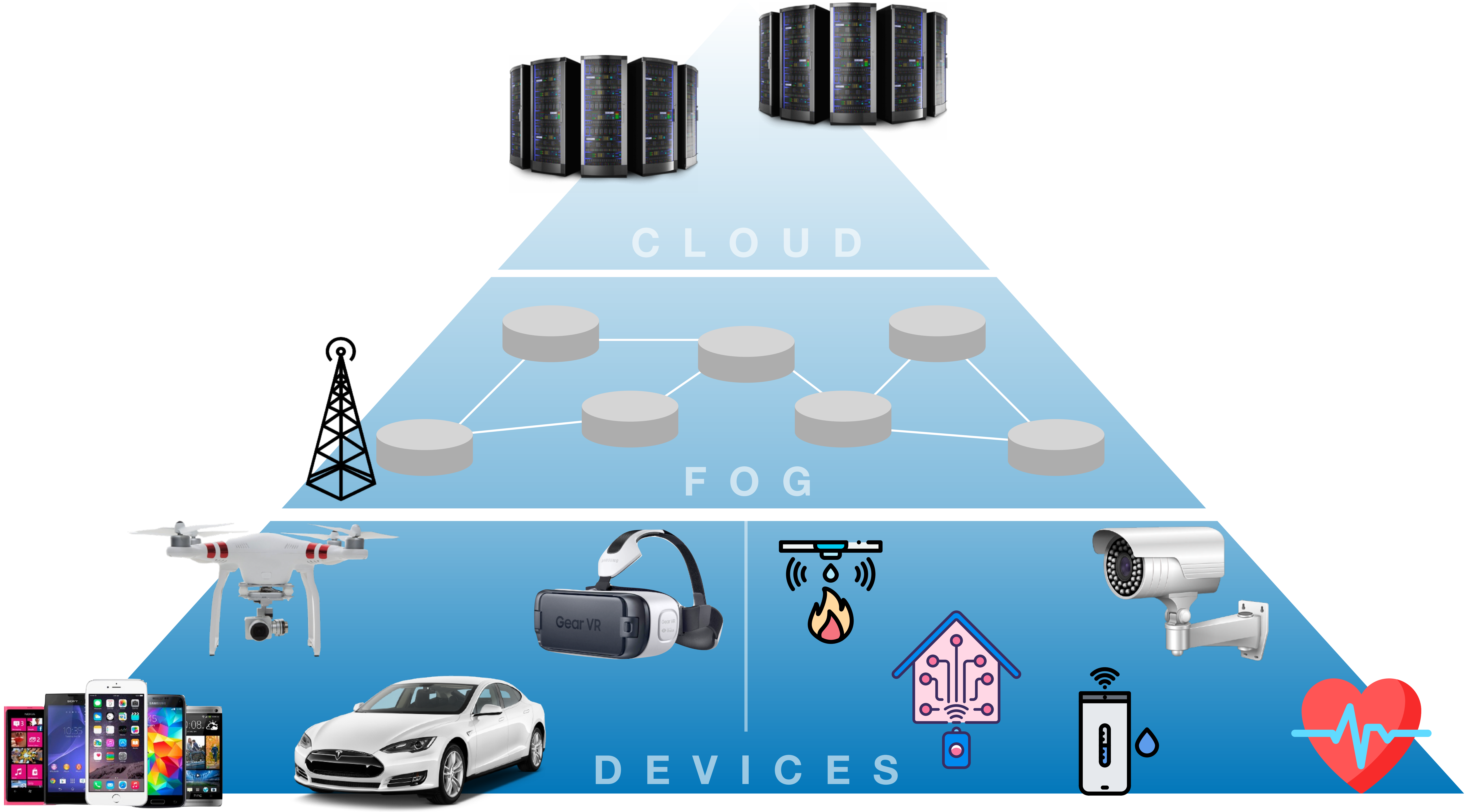
Context and Challenges

Context

Fog architecture

~100ms

Average round-trip-time cloud-to-device

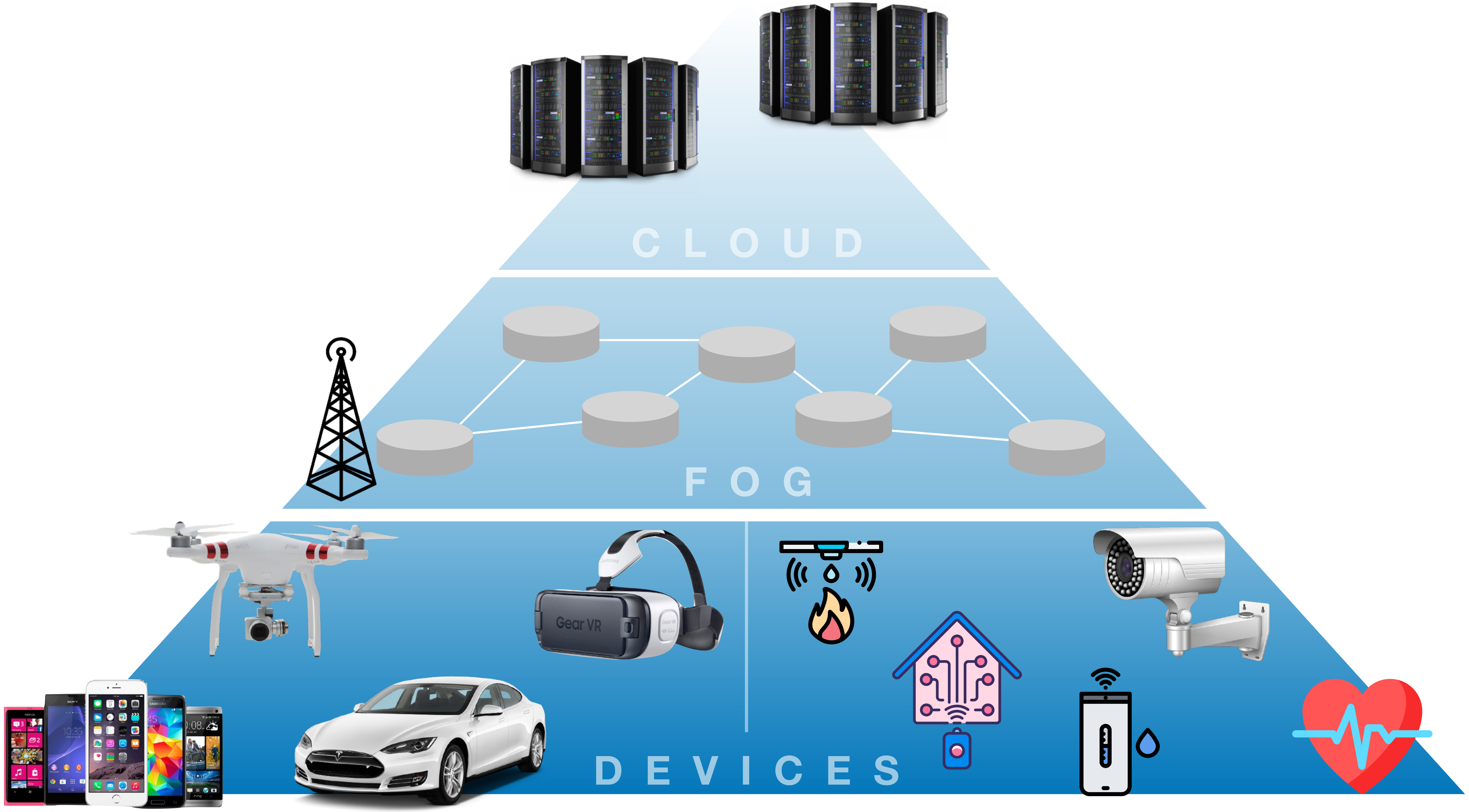


Context

Fog architecture

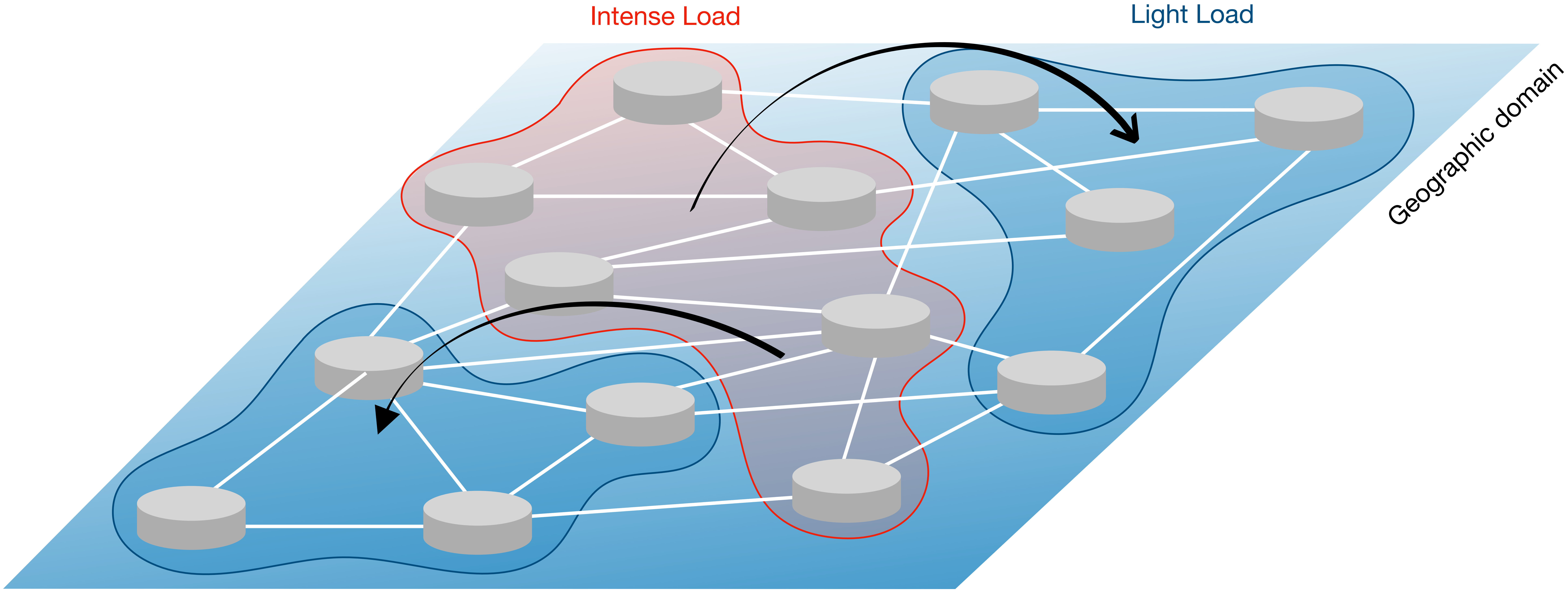
~5ms

Average
round-trip-time
fog-to-device



Challenges

Load balancing

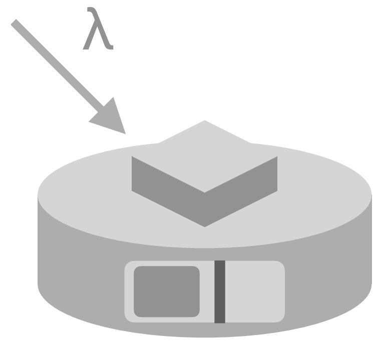


How can nodes cooperate with the lowest overhead?

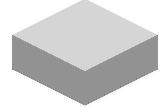
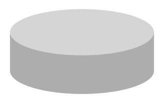
2

Proposed Algorithm

System Model



For modelling the system we assume:

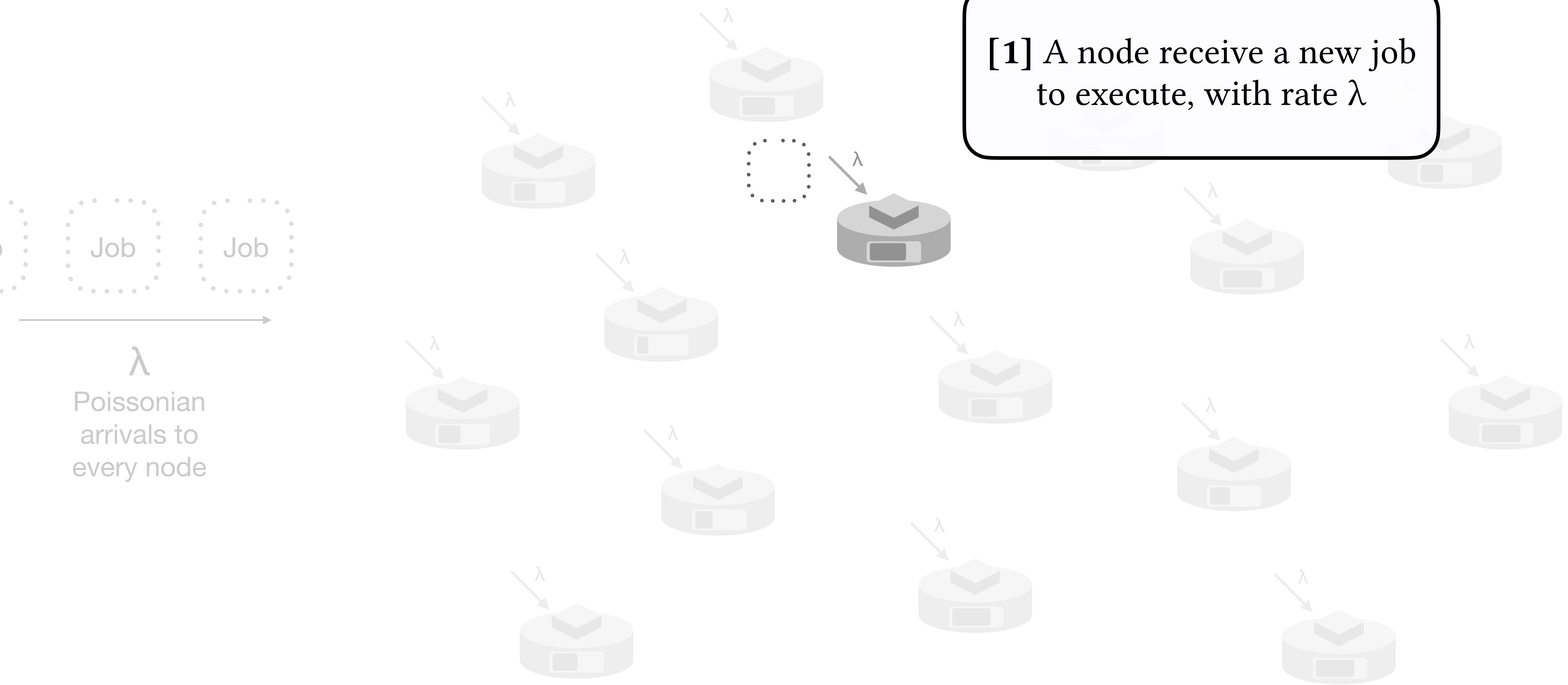
- a set of **N** nodes
- each node is represented as an **M/M/1/K queue**, with max queue length K
- each node receives a **poisson flow** of jobs with rate λ
- **service time** is exponentially distributed with μ = 1
- obviously a node acts both as a scheduler () and as a worker ()

We set:

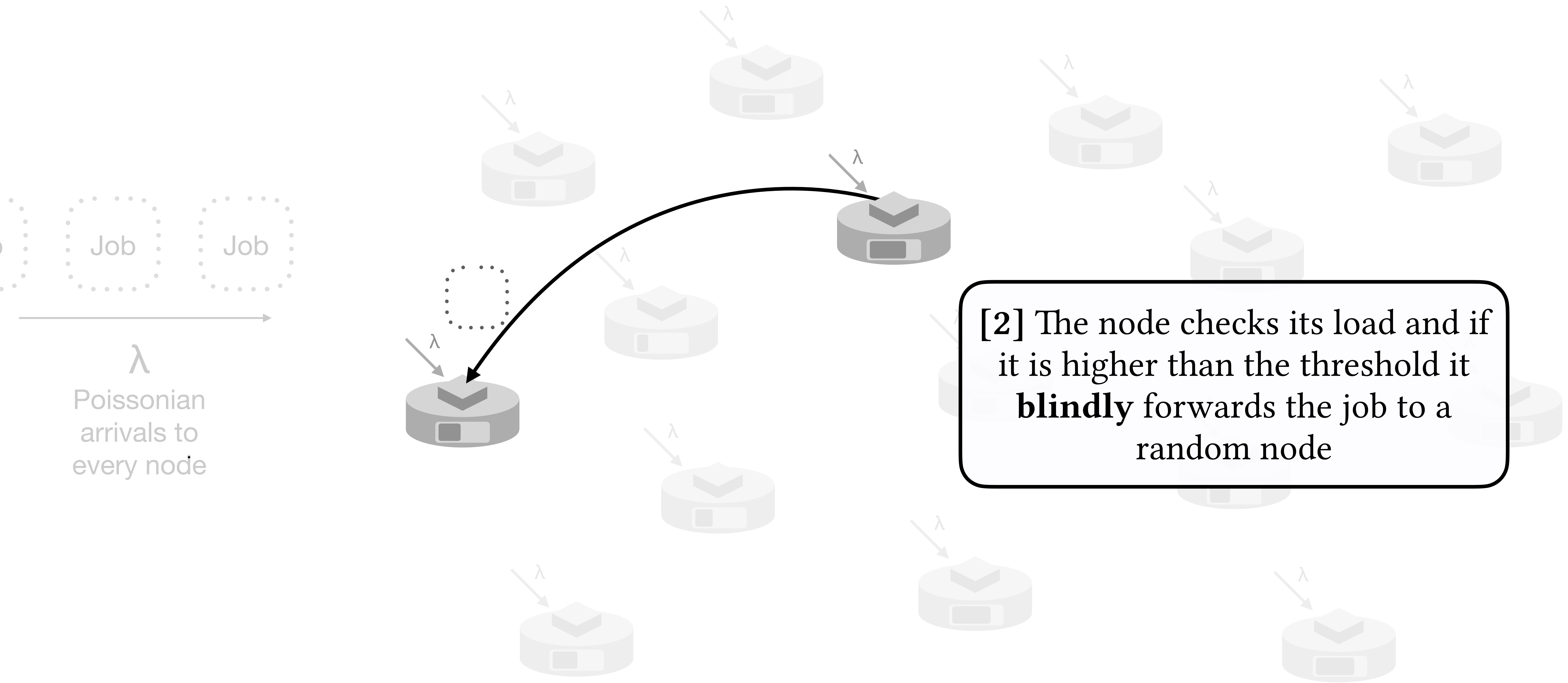
- **M** as the maximum number of **hops** before executing the job
- **Θ** as the cooperating threshold, namely each node starts to **cooperate** only when its load is higher than T

Sequential Forwarding

[1] A node receive a new job to execute, with rate λ

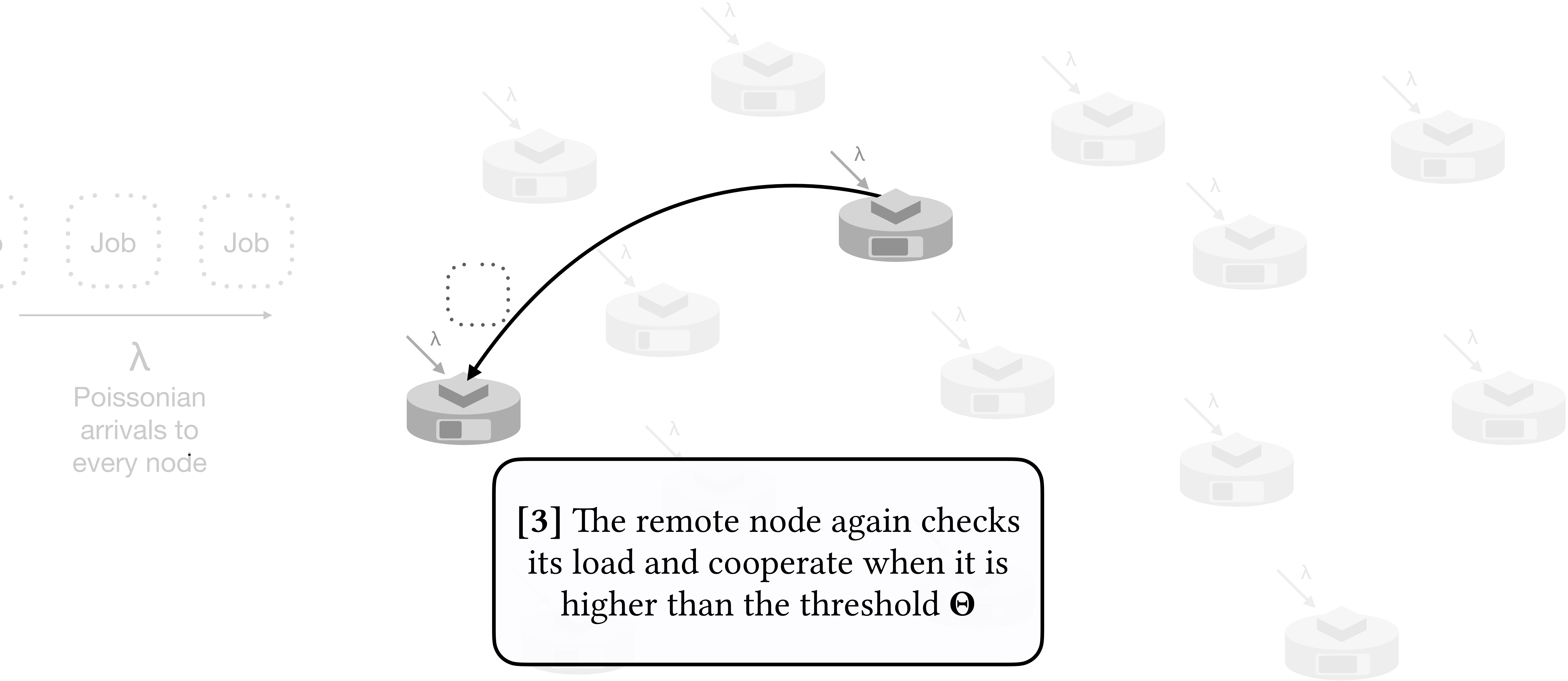


Sequential Forwarding



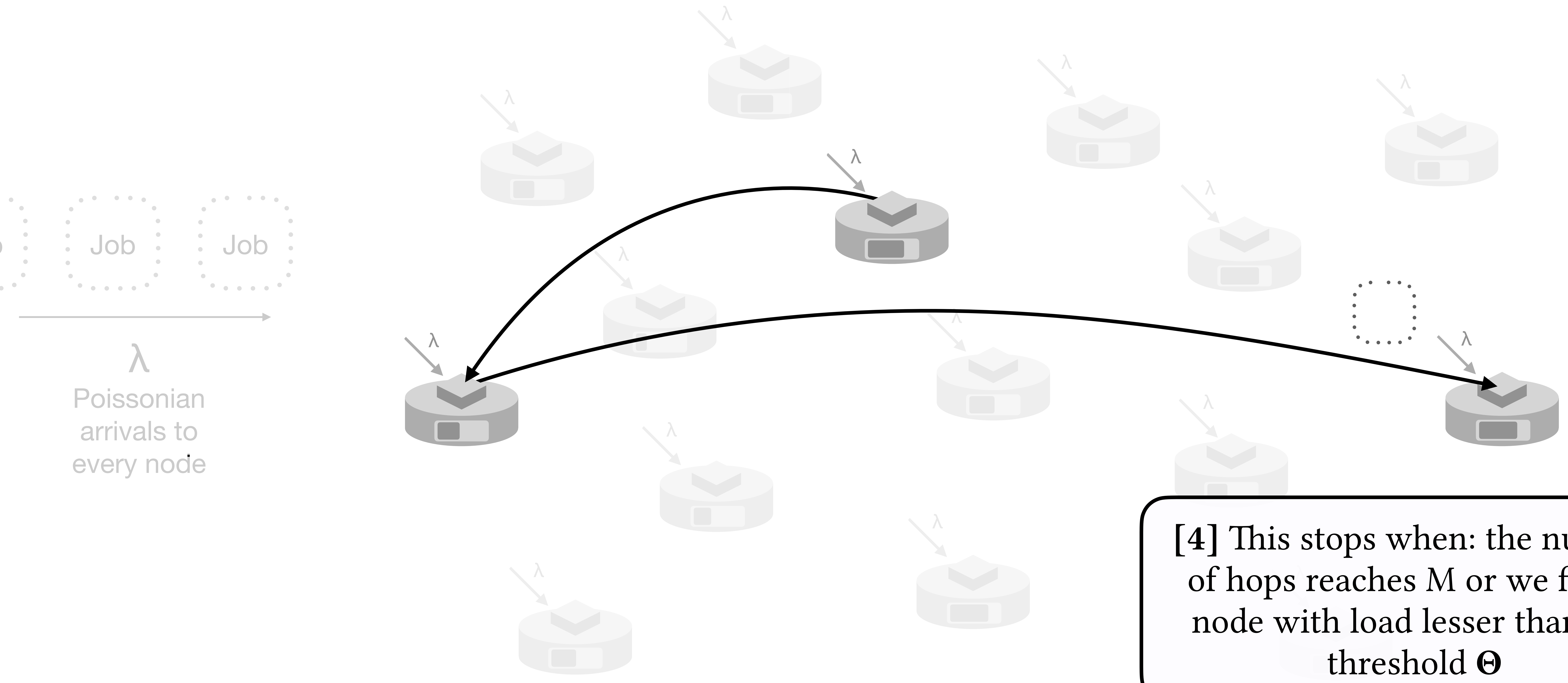
[2] The node checks its load and if it is higher than the threshold it **blindly** forwards the job to a random node

Sequential Forwarding



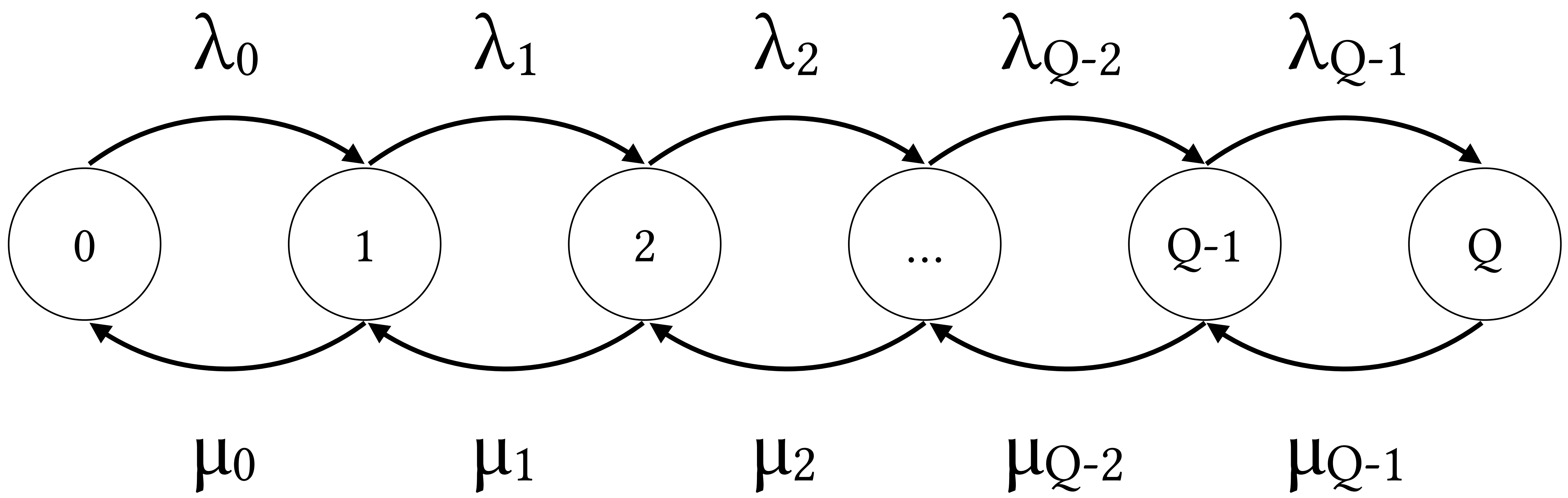
[3] The remote node again checks its load and cooperate when it is higher than the threshold **Θ**

Sequential Forwarding



[4] This stops when: the number of hops reaches M or we find a node with load lesser than the threshold **Θ**

Performance Model



Continuous Time Markov Chain representing a single node (Birth-death process)

$$\lambda_i = \lambda \begin{cases} 1 + \pi_\Theta \sum_{k=0}^{M-1} \pi_\Theta^k = \frac{1 - \pi_\Theta^{M+1}}{1 - \pi_\Theta} & i \leq \Theta \\ \pi_\Theta^M & i > \Theta \end{cases}$$

Birth rates for a node in the state i

Adaptive Sequential Forwarding

The second proposed algorithm is a variation of the already presented one but the threshold value Θ is not fixed but varies at every hop in such a way it grows linearly with the number of steps.

Algorithm 2 Adaptive Forwarding Algorithm

Require: M, Job

$Q \leftarrow$ System.QueueLen()

$\Theta \leftarrow \lceil \text{Job.Steps} * Q/M \rceil$

$\mathit{SequentialForwarding}(M, \Theta, \text{Job})$

Metrics

For understanding the performance of the proposed algorithm we are interested in:

- **S** — average number of steps before executing a job
- $\mathbf{W_Q}$ — average queue length
- $\mathbf{p_B}$ — *blocking probability* or *loss rate*, namely the percentage of job that cannot be served
- $\mathbf{T_r}$ — average total service time
- $\mathscr{J}$ — fairness, from the Jain index

We both provided numerical results and simulation results of the proposed load balancing strategy.

3

Results

Simulations Setting

Simulations have done both with a *simplified scenario* in which we evaluated the model and with a *realistic scenario* which reflects a true deployment of 20 fog nodes and 100 sensors in the city of Modena, Italy.

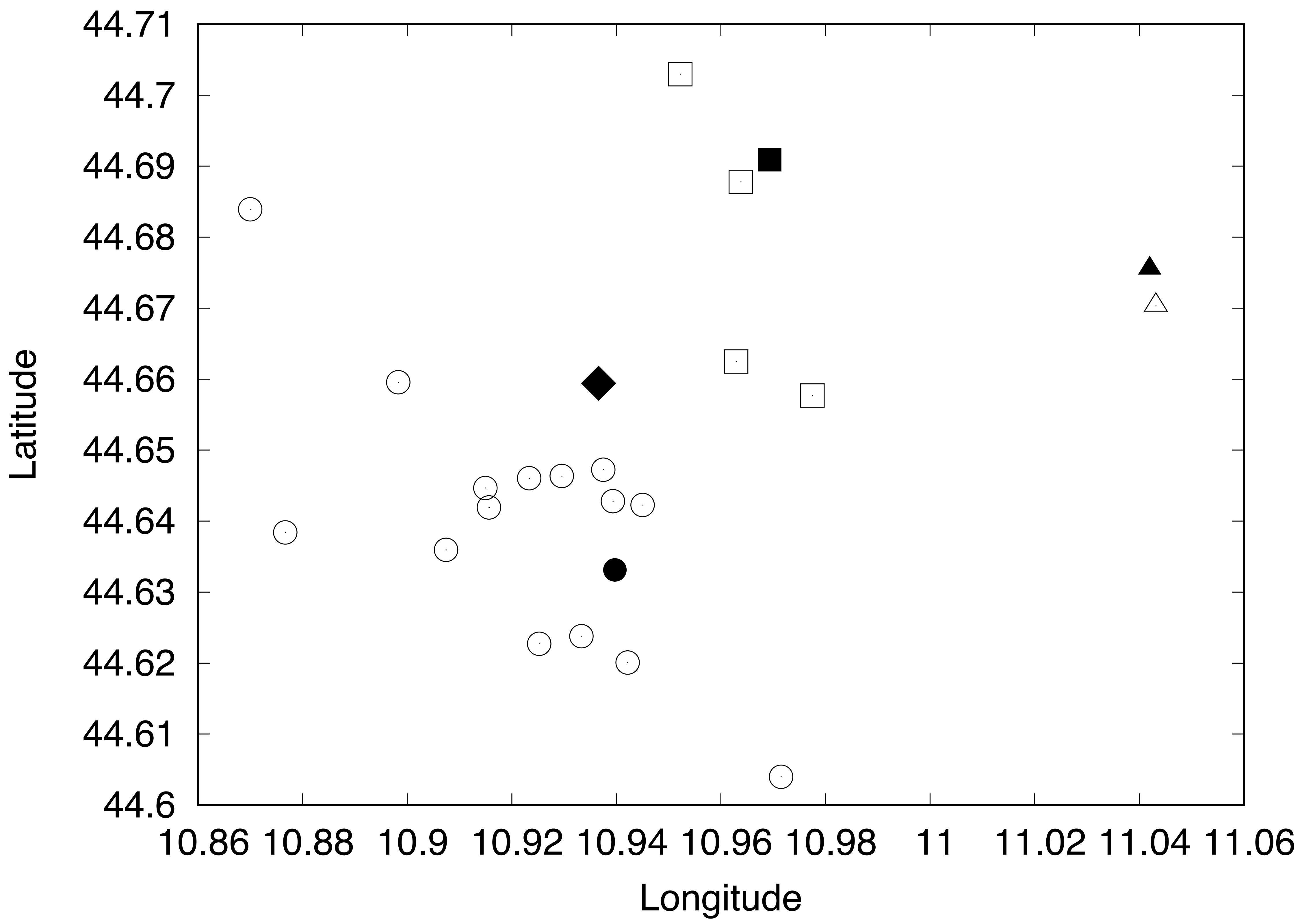
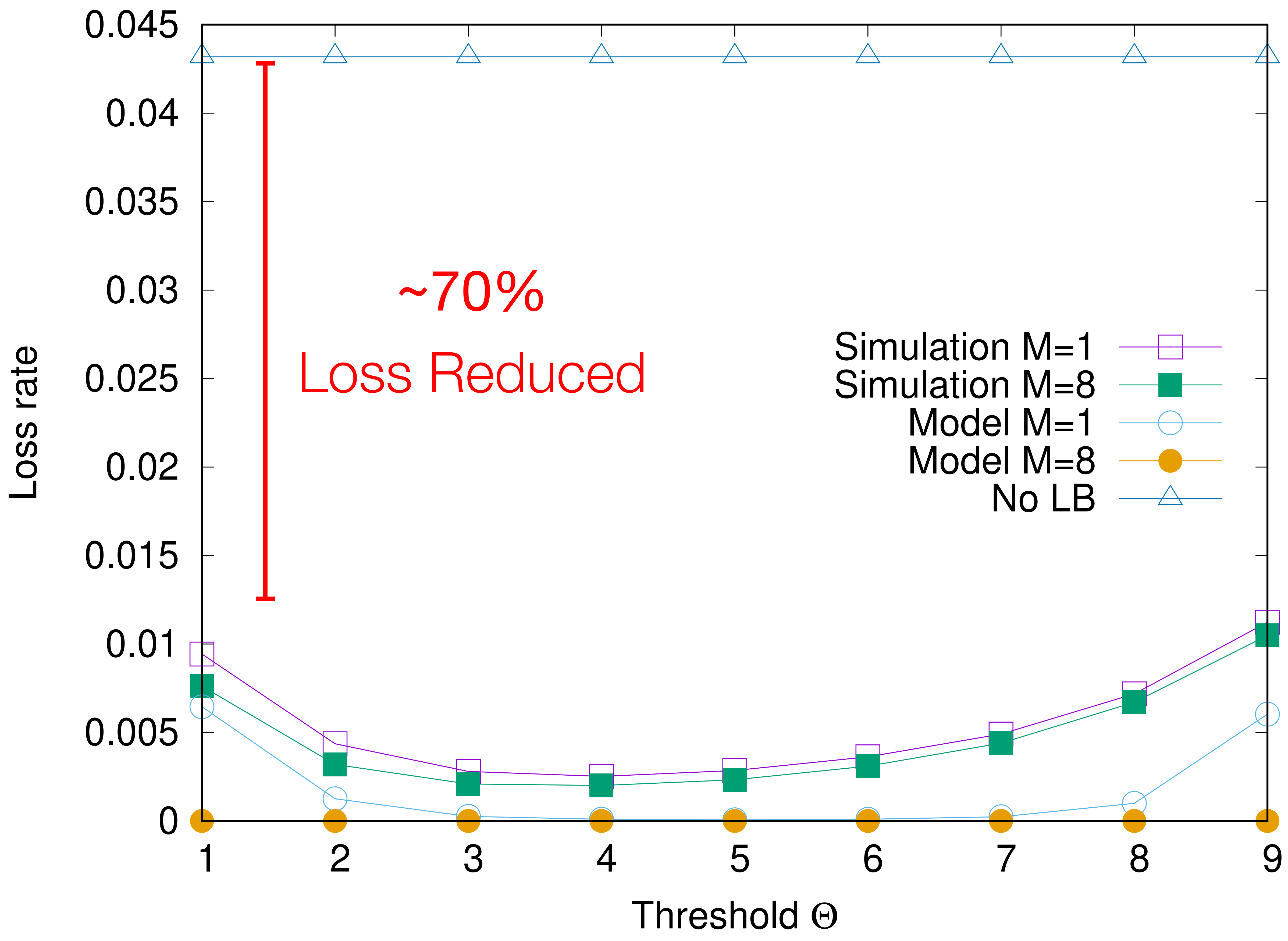
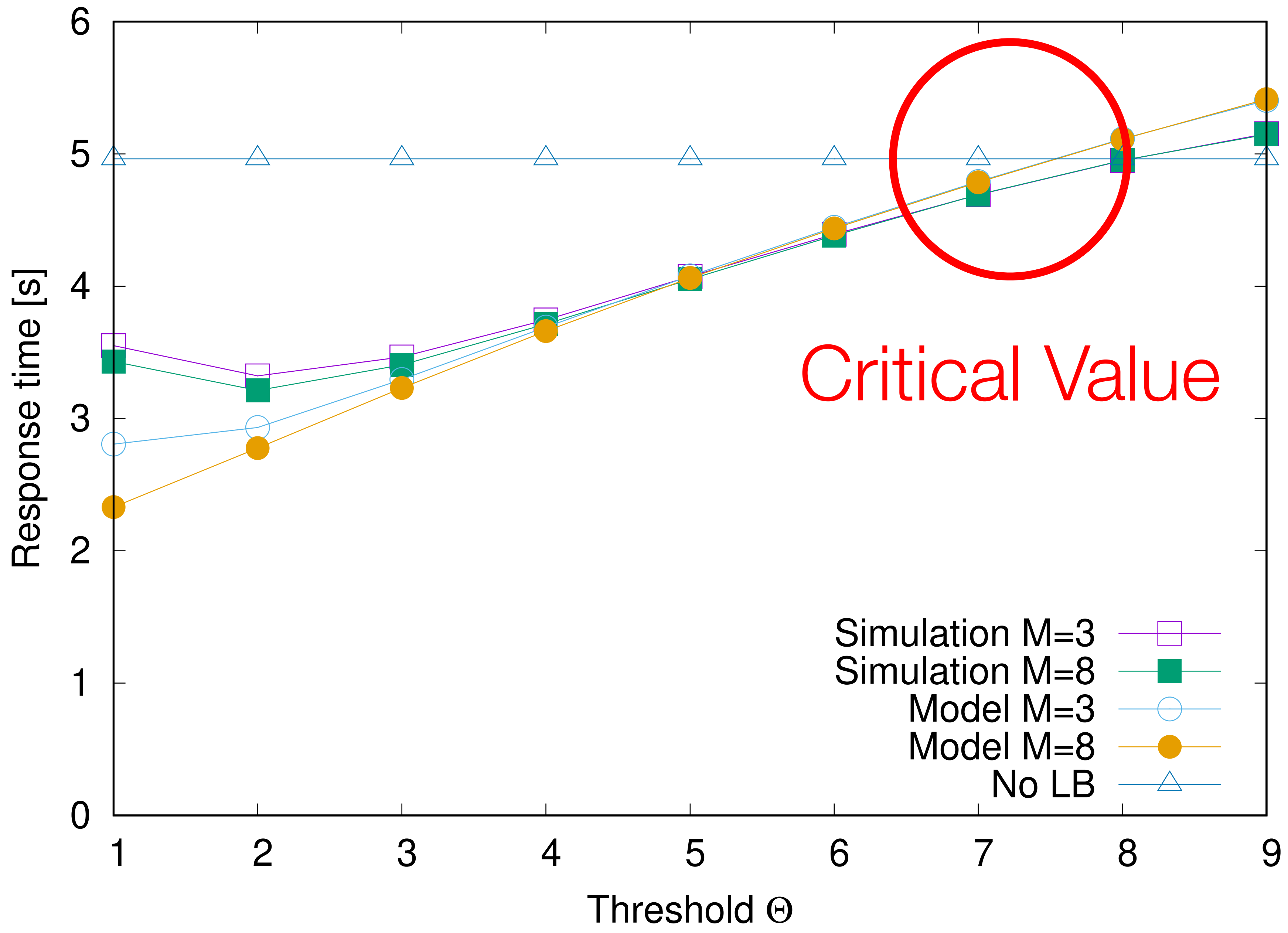


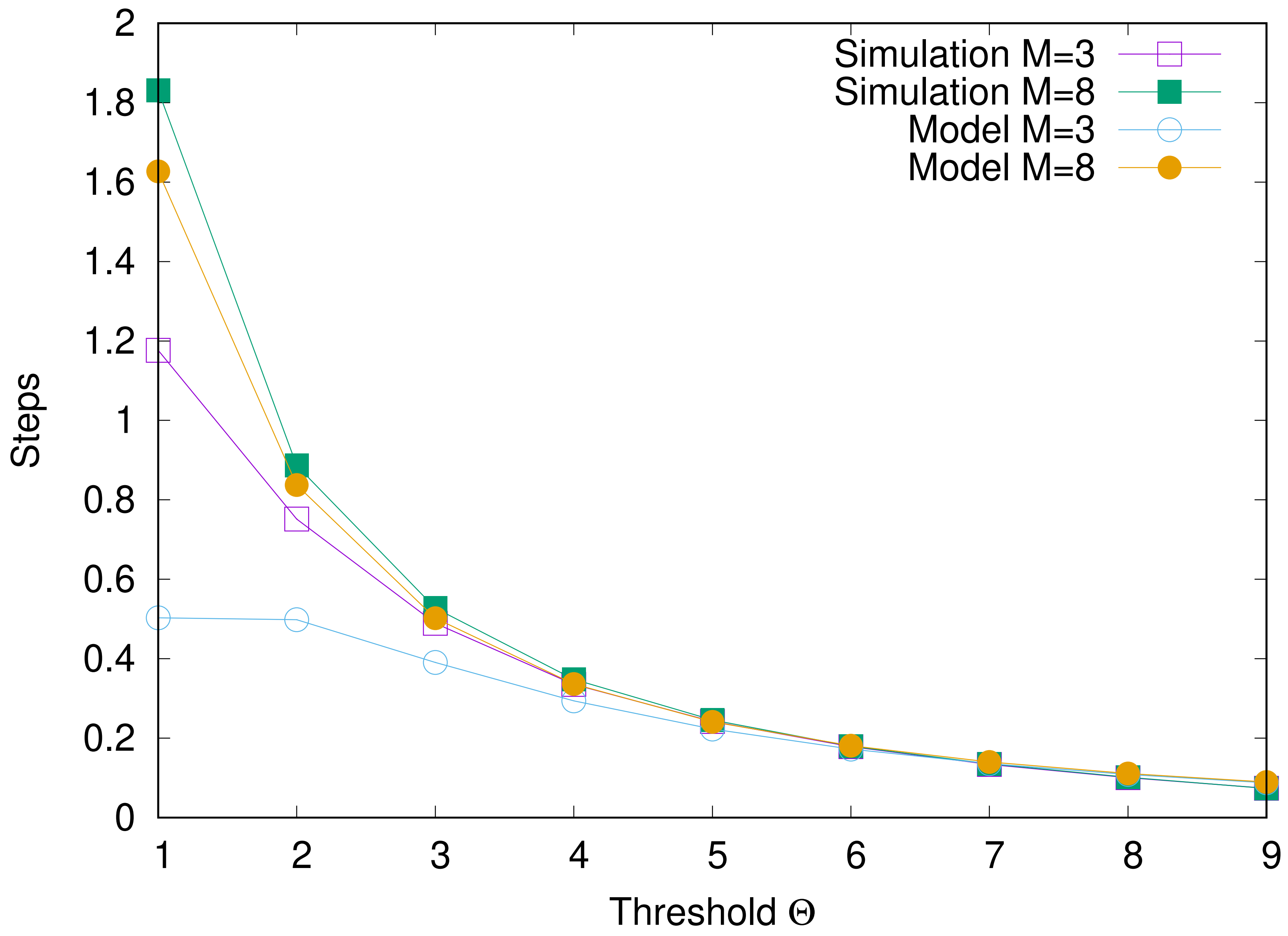
Figure. Placement of Fog nodes in the *realistic scenario* simulations.
Here we assume that the time needed for two nodes to communicate is proportional to their distance

Results

Loss Rate, Response Time and Steps

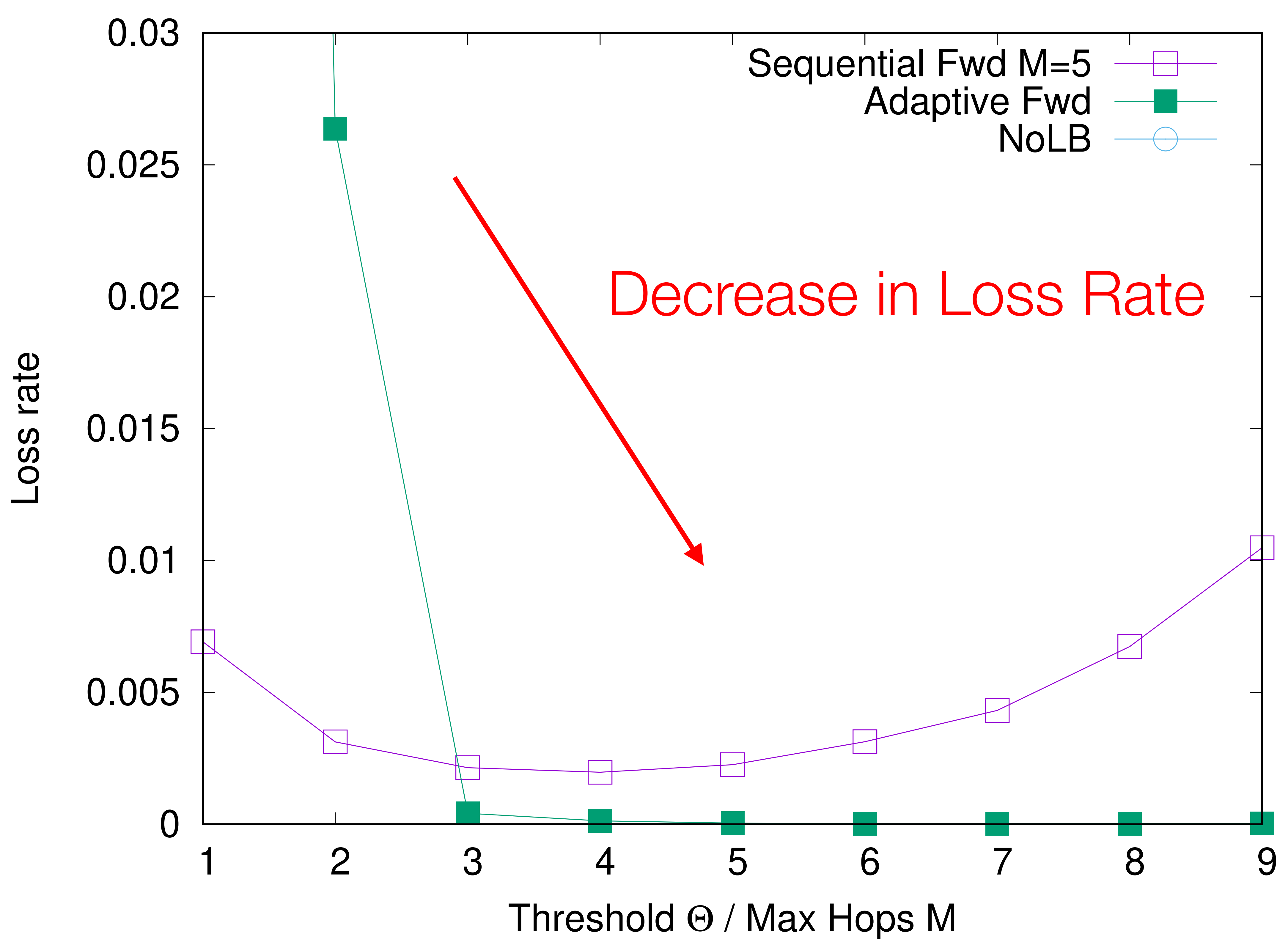


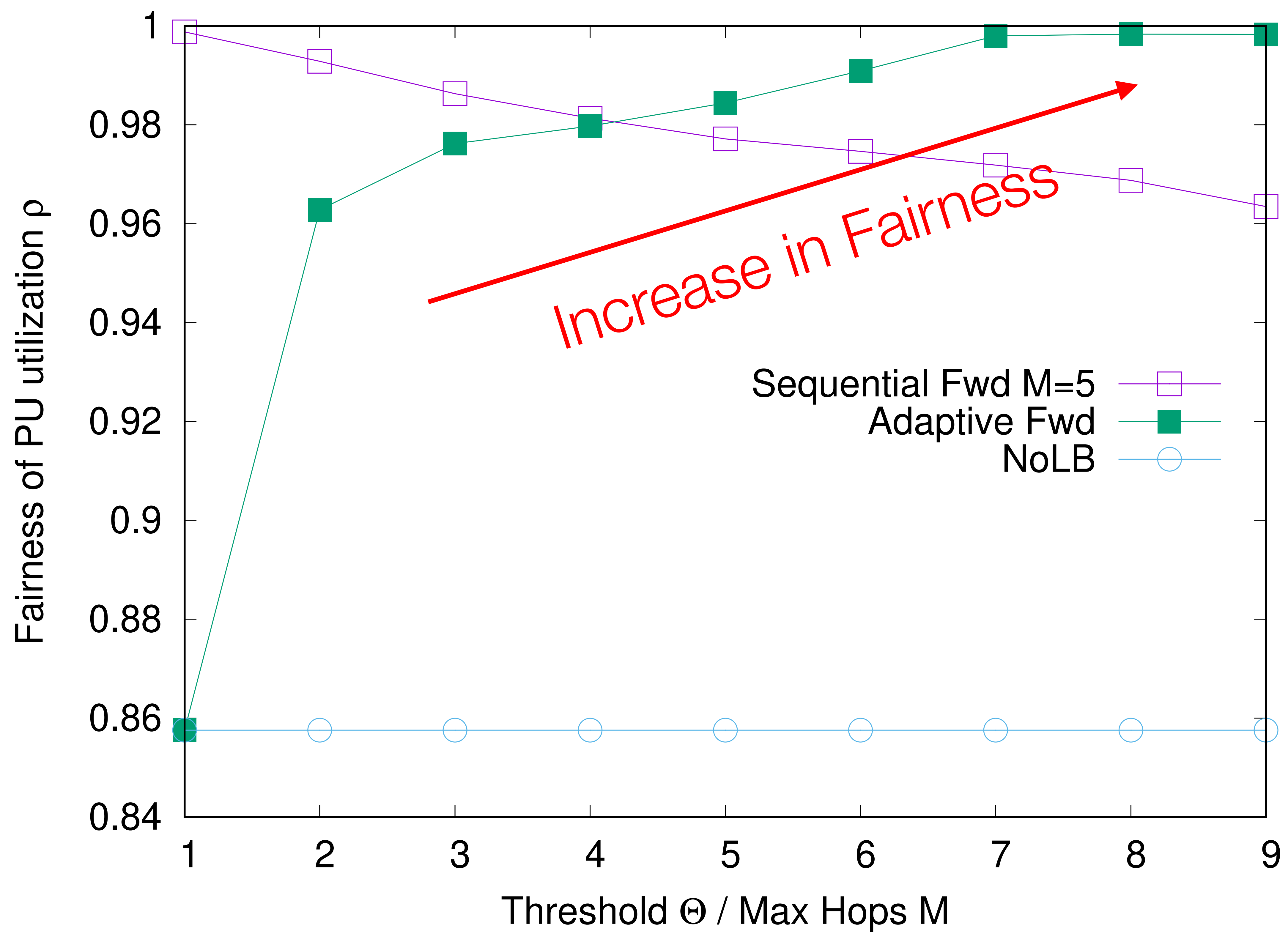




Results

Adaptive Sequential Forwarding





4

Conclusions

Conclusions

- we proposed two **algorithms**, namely *sequential forwarding* and the *adaptive sequential forwarding* for **load balancing** in a Fog environment
- we tested our algorithms both with a **mathematical model** and with **simulations**
- results shows that in the *simplified scenario* the model **is validated** against the simulations
- in the *realistic scenario* the algorithms **outperforms** the case in which no load balancing is adopted: loss rate drops from 25% to 0.5% for the seq. for. and 0% for the adaptive
- the *adaptive sequential forwarding*, with its threshold **self-tuning** mechanism can provide **stable performances** in terms of low response time and low loss rate, also guaranteeing fairness (Jain index close to 1)

A random walk based load balancing algorithm for Fog Computing

Roberto Beraldi[*], Claudia Canali[×], Riccardo Lancellotti[×],
Gabriele Proietti Mattia[*]

PRESENTATION
Gabriele **Proietti Mattia**

[*]Department of Computer, Control and Management Engineering "Antonio Ruberti", Sapienza University of Rome, Italy
[×]Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, Modena, Italy