UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Randomized load balancing under Loosely correlated state information In fog commuting

Roberto Beraldi
“La Sapienza” University, Rome

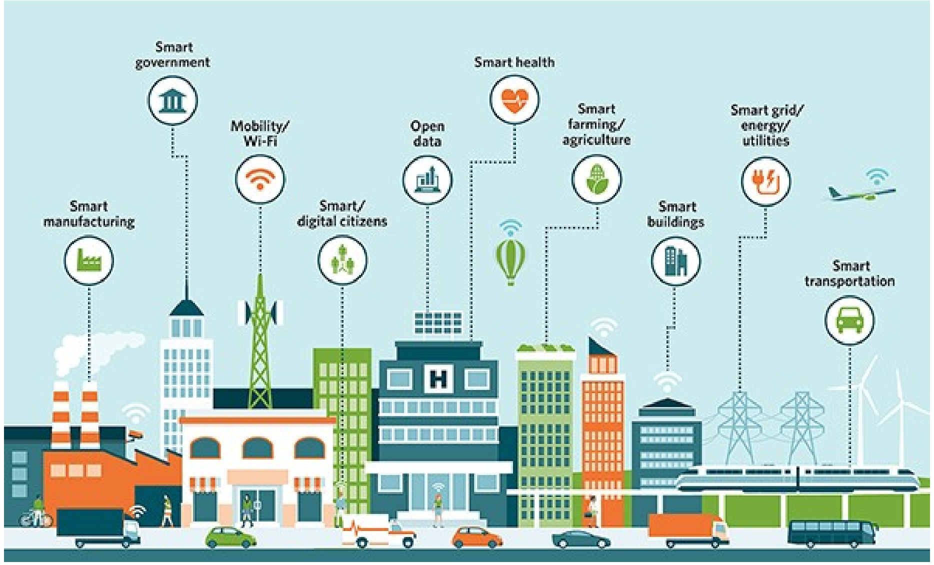
Claudia Canali,
Riccardo Lancellotti
University of Modena and Reggio Emilia

Gabriele Proietti Mattia
“La Sapienza” University, Rome

Motivation for Fog computing

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

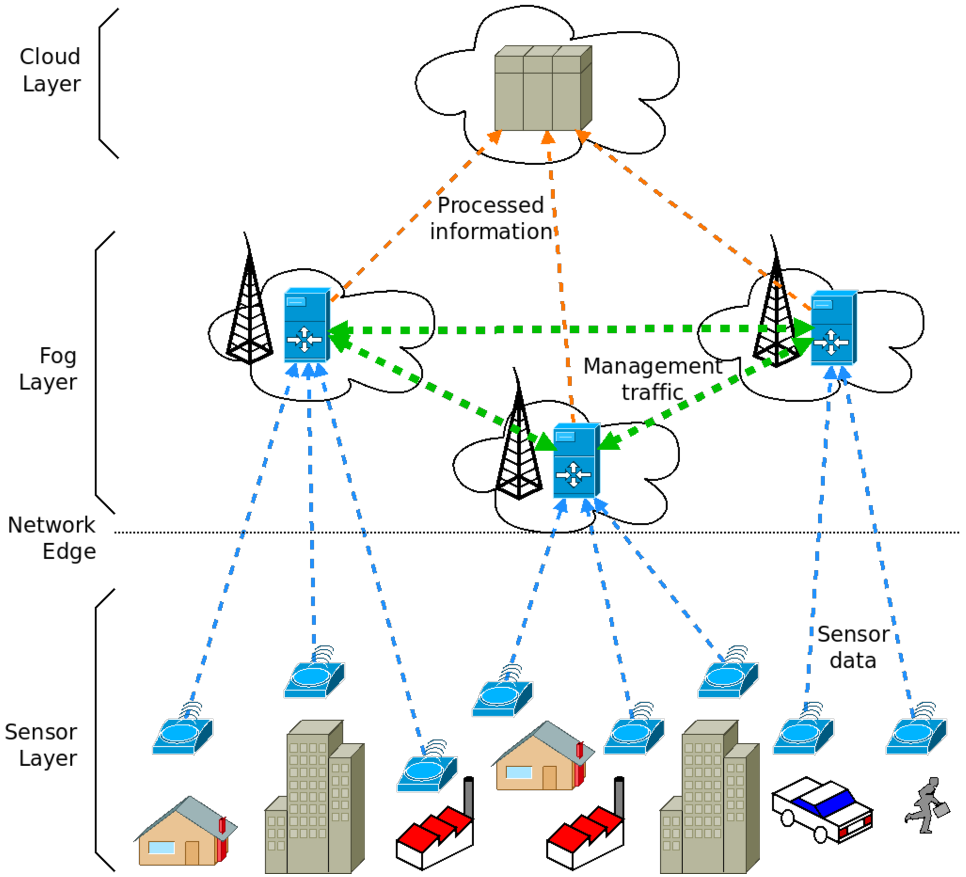
- New paradigm: Smart cities, large scale sensing applications
- Several fields of application:
  - Urban applications
  - Industrial
  - Automotive
  - Healthcare
  - ...



- New scenarios: Cyber-physical systems
  - Geographically distributed sensors
  - Huge amount of data produced
  - Data processing (aggregation, filtering, …) close to sensors

Fog architecture overview

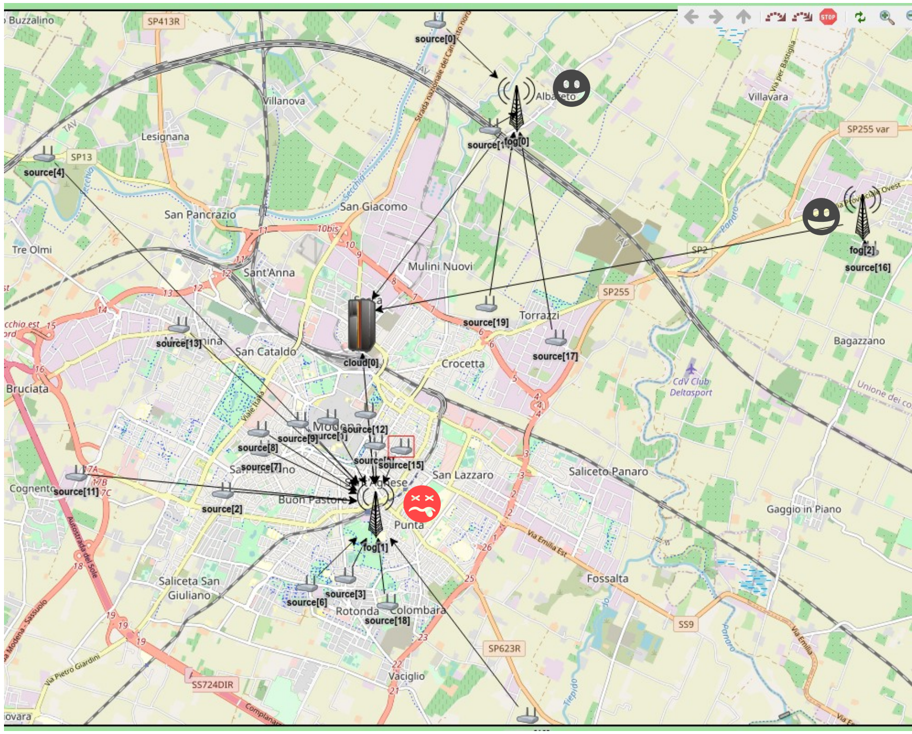
- Sensors: data production
- Fog: data processing
  - Aggregation, filtering
  - Latency-critical tasks
- Cloud: complex applications
- Performance factors:
  - Network delay
  - Processing time @ fog node
- Addressing delay: optimize infrastructure topology
- Still need to manage the infrastructure
  - → Load balancing



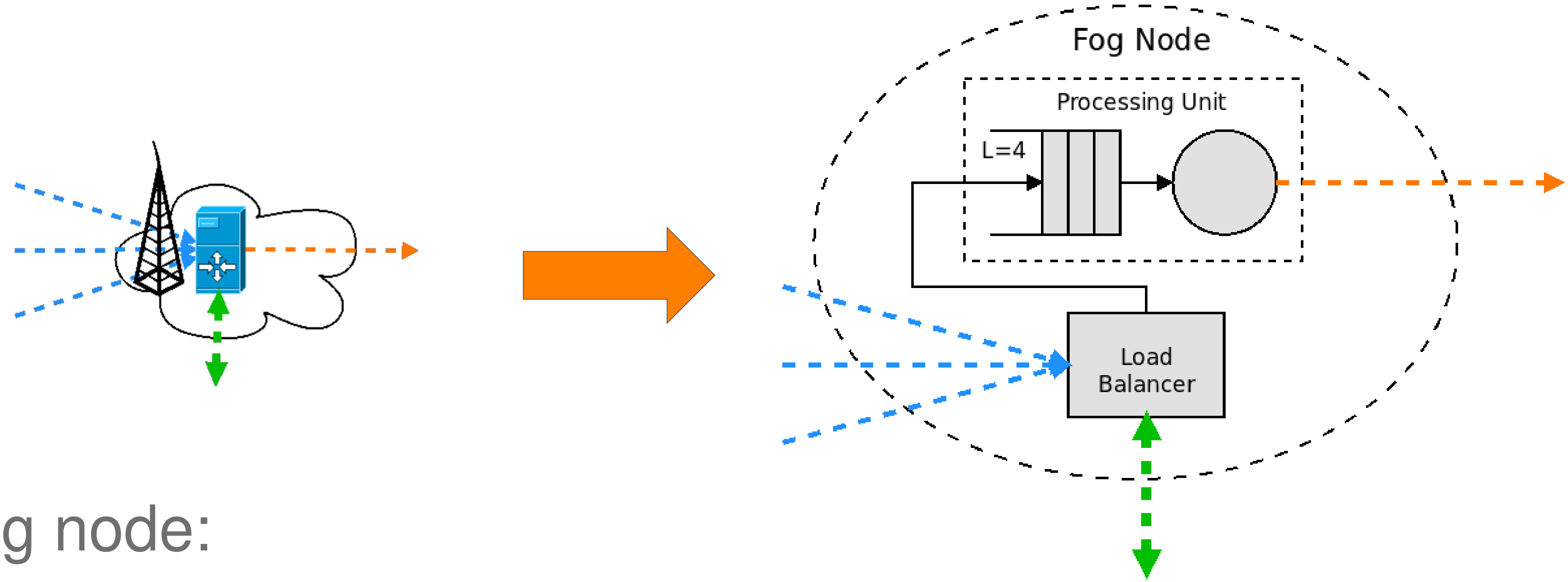
Load balancing in Fog

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

- Two reasons for load balancing
  - Unbalanced incoming load
  - Transient load fluctuations in the infrastructure
- Forward jobs
  - From overloaded nodes
  - To underloaded nodes
- Requires:
  - Knowledge of local load
  - Interaction with other nodes



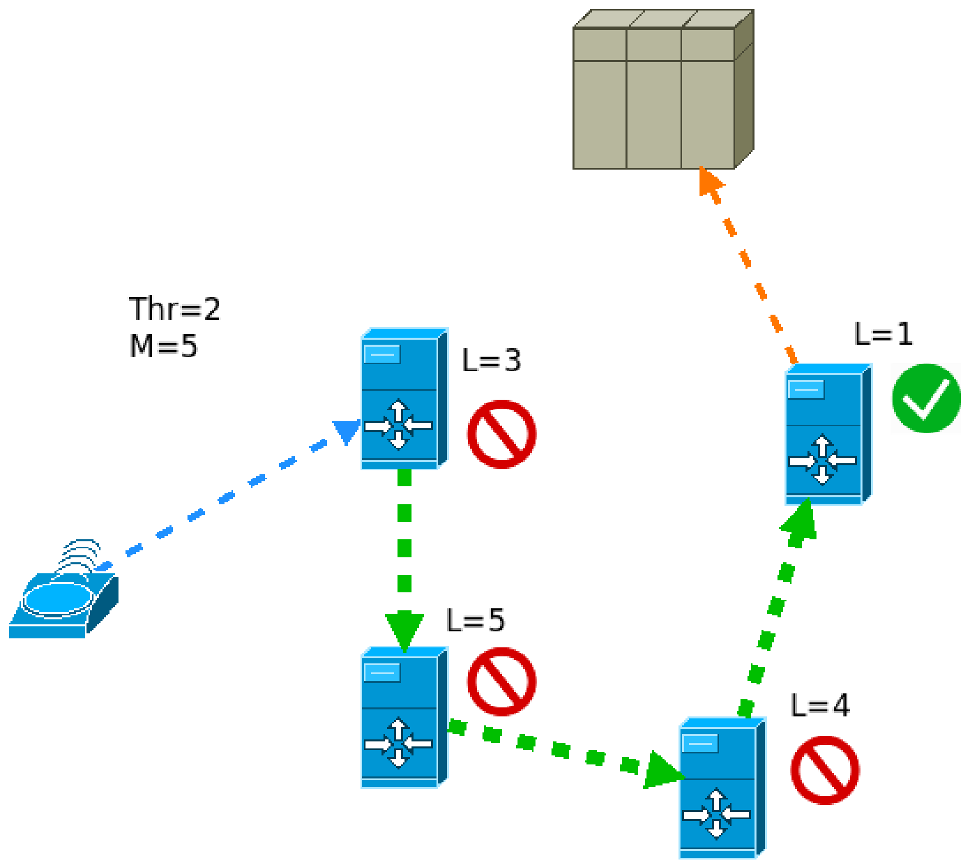
Detail of a fog node



- Fog node:
  - Load balancer (LB – implements balancing algorithm)
  - Processing unit (PU – server with queue)
- Load: jobs in processing unit
- Finite queue size: PU can drop jobs

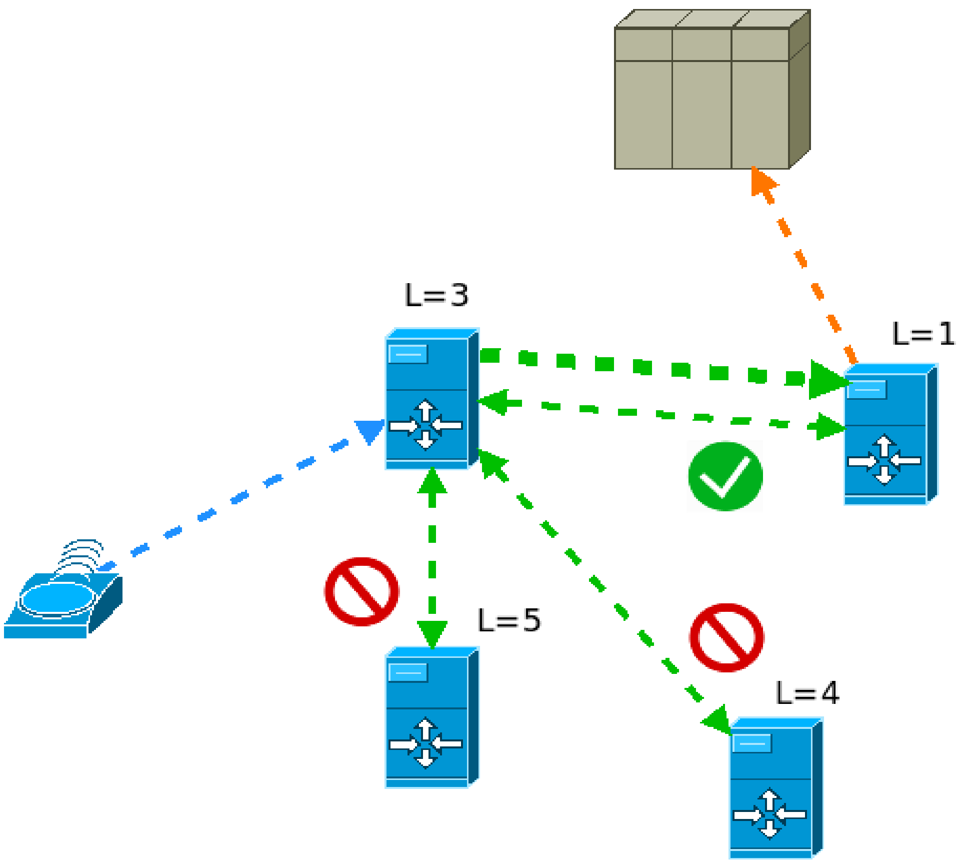
Load-blind approach

- Limited knowledge of load
  - Local load: OK
  - Remote load: NO
- Algorithm:
  - Use of threshold to determine load overload
  - If overloaded randomly forwards to neighbor
  - Up to M hops
- Potentially inefficient
- Extremely fast and simple
- Needs parameter tuning (Thr)



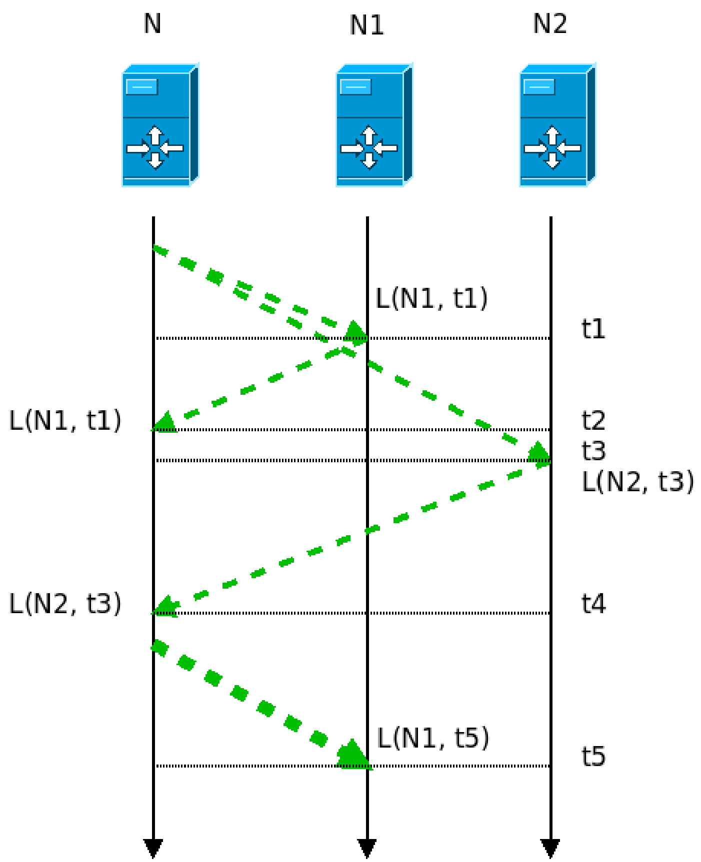
Load-aware approach

- Knows of local and remote load
- Algorithm:
  - If local overload
  - Probe load from neighbors
    - Query message
    - Response message
  - Select least loaded
  - Forwards to selected neighbor
    (or process locally)
- Potentially more efficient
- Risk from inaccurate info



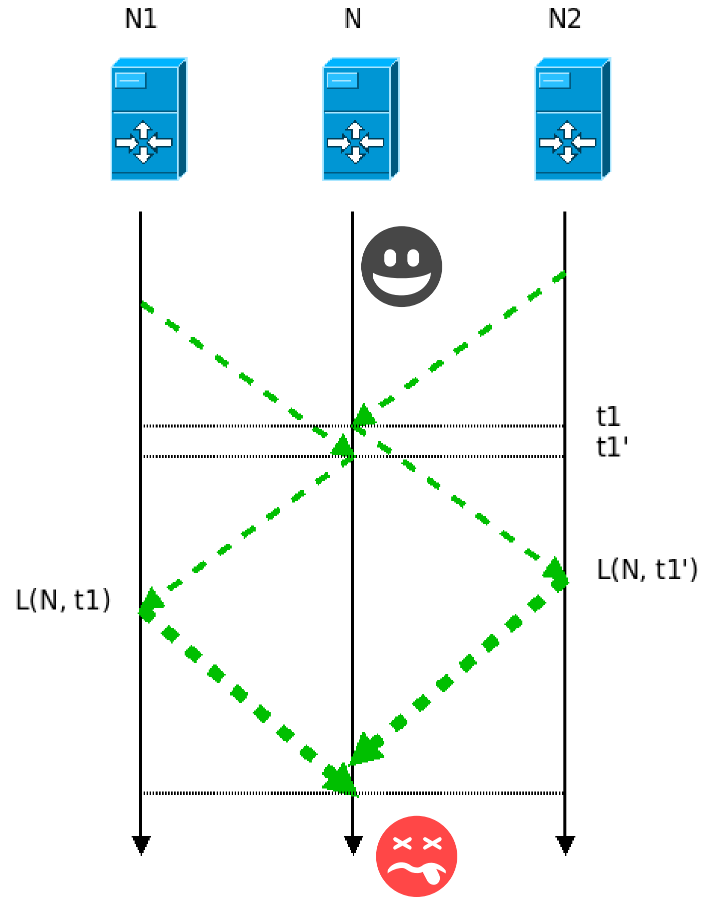
Stale load information

- Stale load:
  - @t2 we know L(N1, t1)
  - @t4 we know L(N2, t3)
  - we select N1 based on stale load information
  - @t5 L(N1, t5) $\neq$ L(N1, t1)
- Load changes over time
- Communication delay grows
  - Inaccurate information
  - Probing less useful



Herding effect

- Two neighbors probe node N
  - N is underloaded
  - All neighbors select N
  - N → overload
- Delay in information propagation
  - Effect of stale load
  - Can cause oscillations
  - Occurs when a node receives multiple concurrent probes

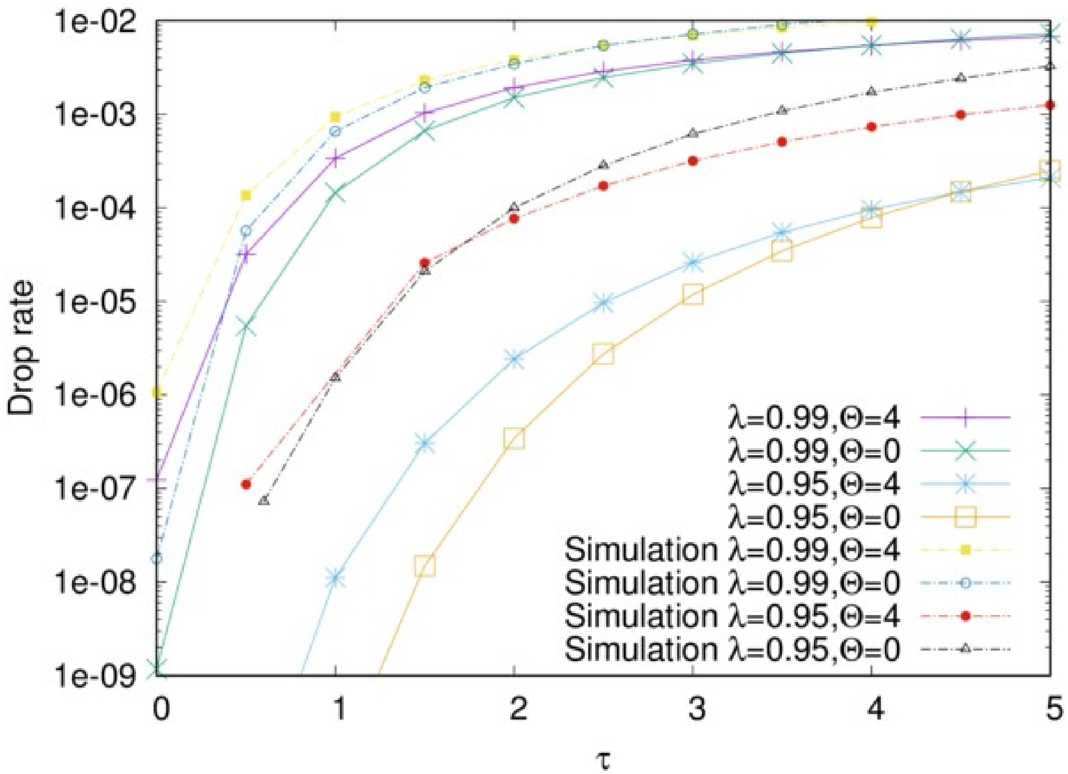


Stale information model

- Queuing model
  - Processing unit
  - Message queue for data exchange among nodes
- Model loss of correlation between:
  - Actual load
  - Advertised load
- Perfect correlation → ideal case
- Complete loss of correlation → random load information
- Correlation depends on:
  - Network delay
  - Network load (affects delay)
  - Load evolution dynamic

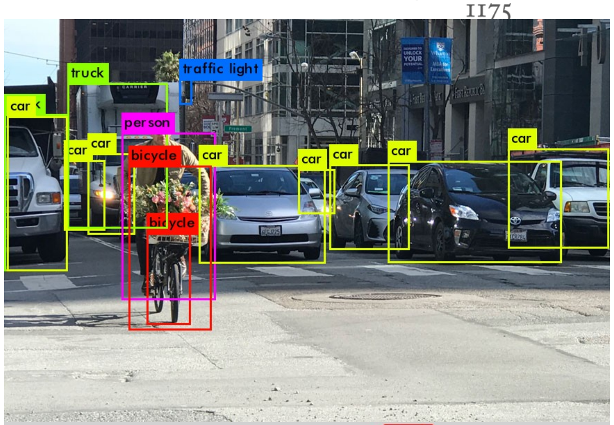
Numerical results

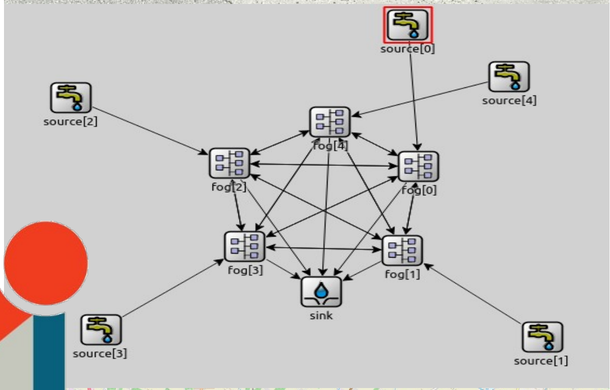
- Drop rate vs. Delay
- Scenario
  - Incoming load
  - Threshold
- Delay grows:
  - Increase of drop rate
- Effects of scenario
  - High load → faster degradation
  - High threshold → more aggressive probing
  - Detrimental effect for high delay

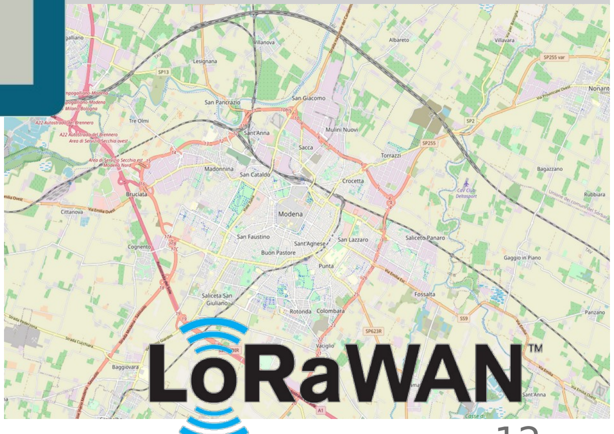


Simulation setup

- Traffic monitoring application, OMNET++ sim
  - Processing images from camera
  - Object identification (cars/humans/bikes)
  - Limited queue (K=10, RT application)
  - Proc. time ≈ Network delay (10 ms)
- Two scenarios:
  - Uniform mesh
    - Same load for fog nodes
    - Same BW in communication
  - Geographic scenario (Modena)
    - Highly variable load
    - Based on real topography
    - BW $\propto$ 1/distance (LoRaWAN)



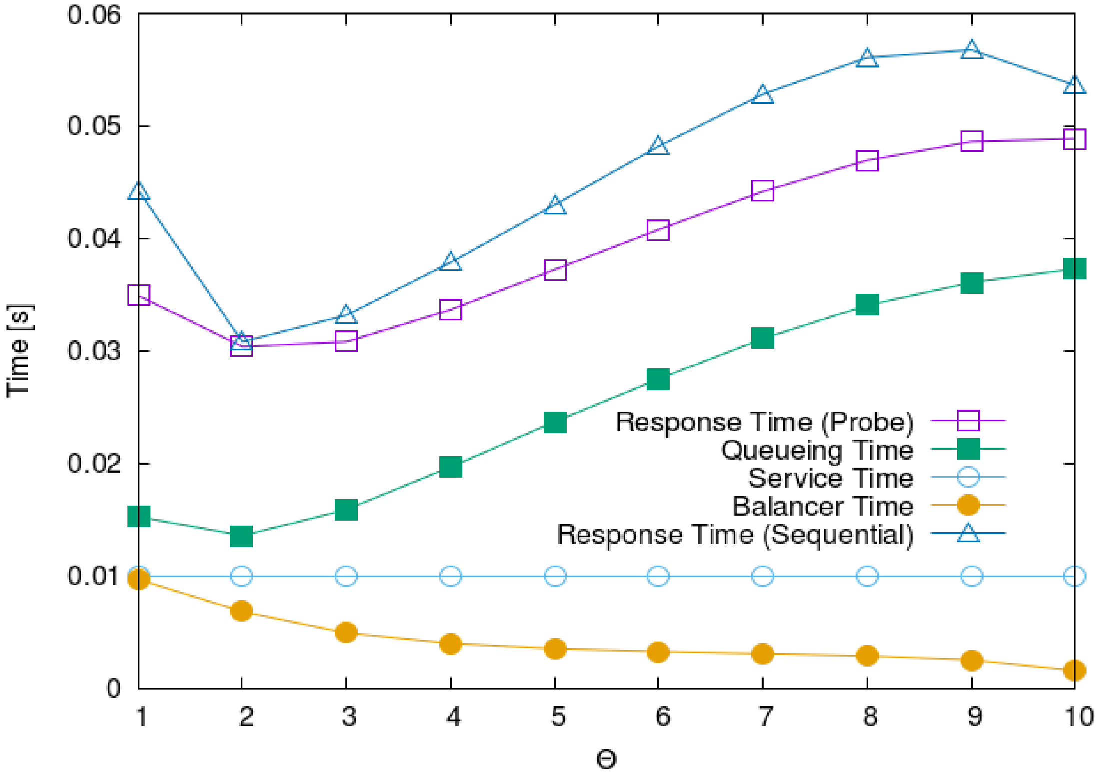




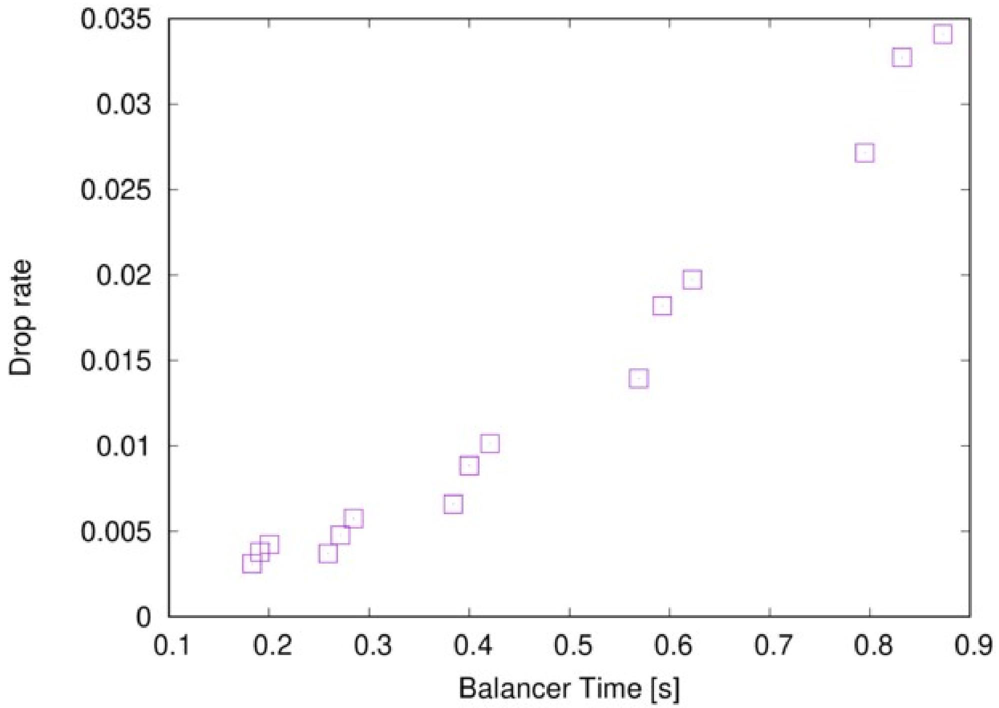
Simulation results

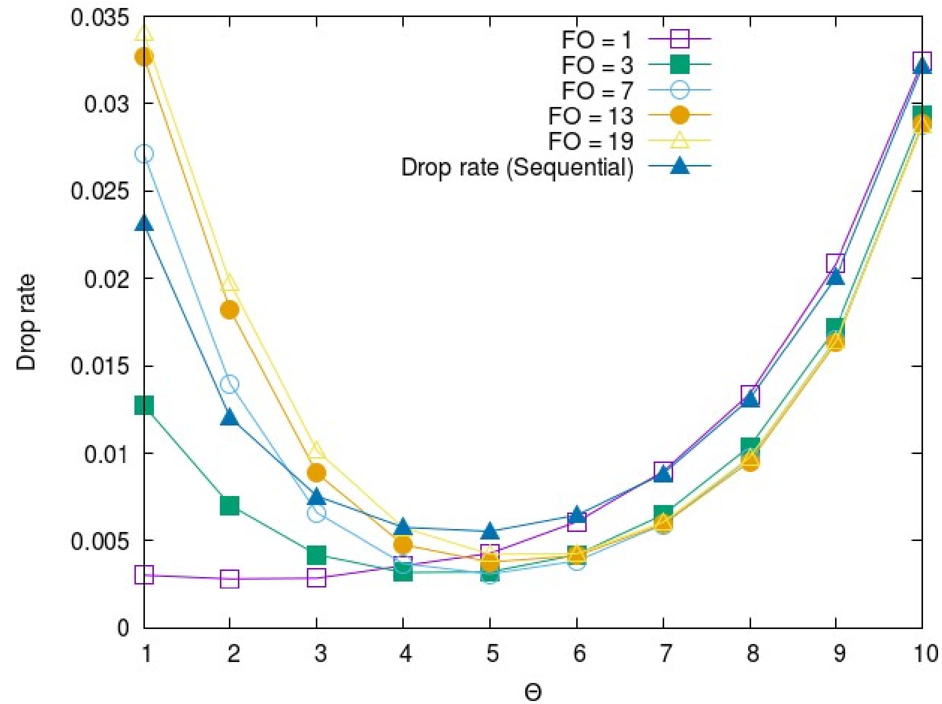
UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

- Mesh scenario
  - Times *vs.* Threshold Θ
- Queuing time
  - Increases with Θ
  - Less balancing → longer queues
- Balancer time
  - Decreases with time
  - Less invocations
  - Less network load
- Response time
  - Sweet point for Θ=2
  - Probe faster (and more stable) than Sequential



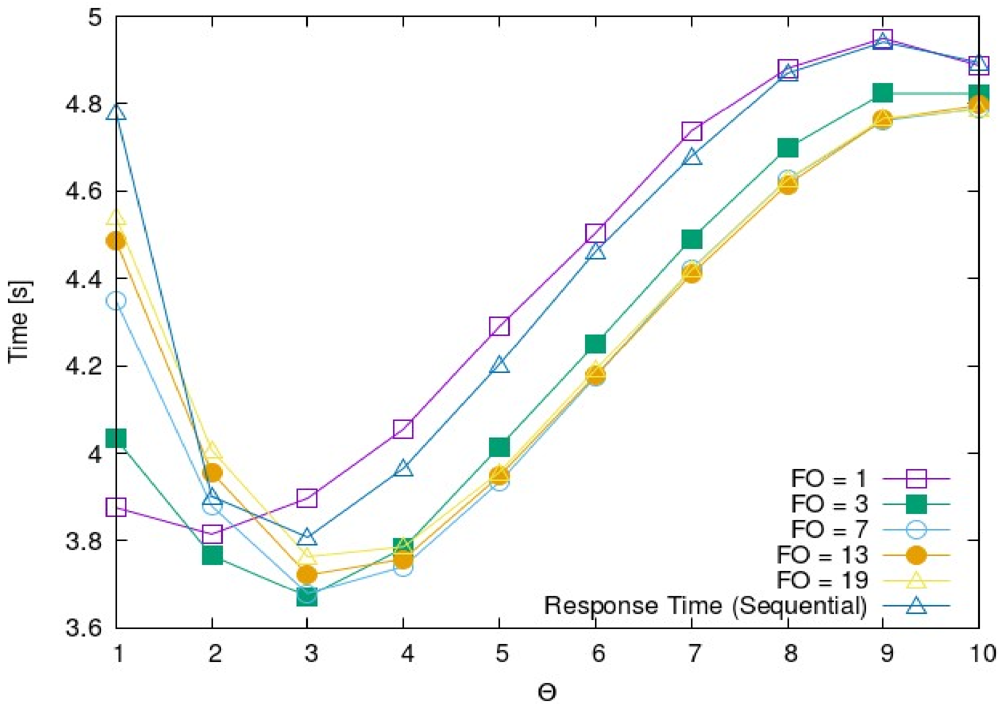
Mesh scenario

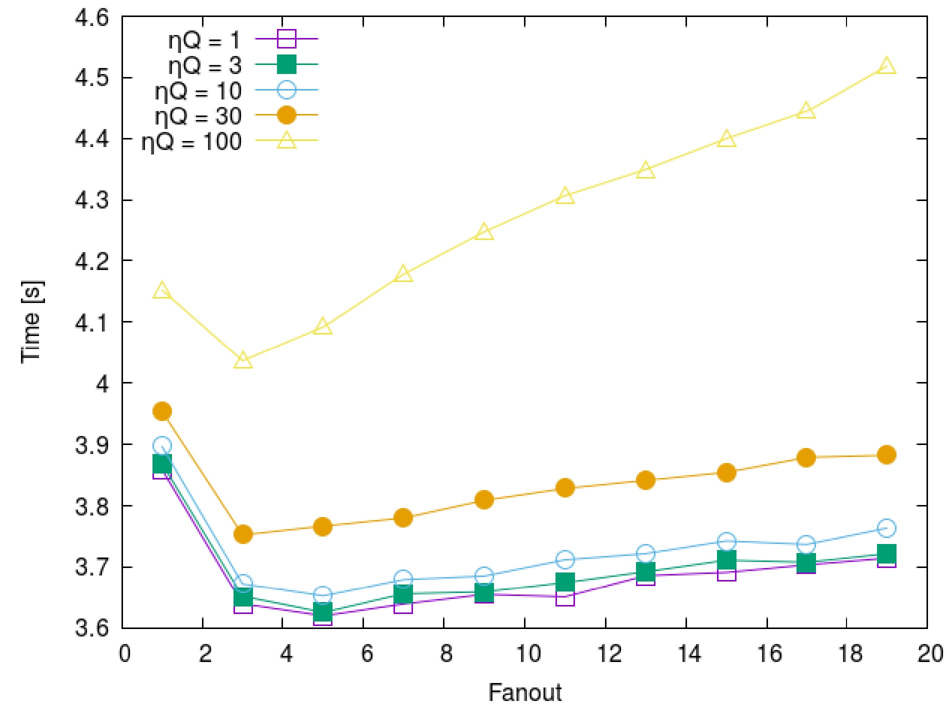




- Strong correlation Drop rate / Probe delay (Balancer time)
- Cup shaped curve of drop rate *vs.* Θ
- Herding effect:
  - High Fan out, many queries → Like load-blind (or worse!)
  - Low fan out reduces this effect

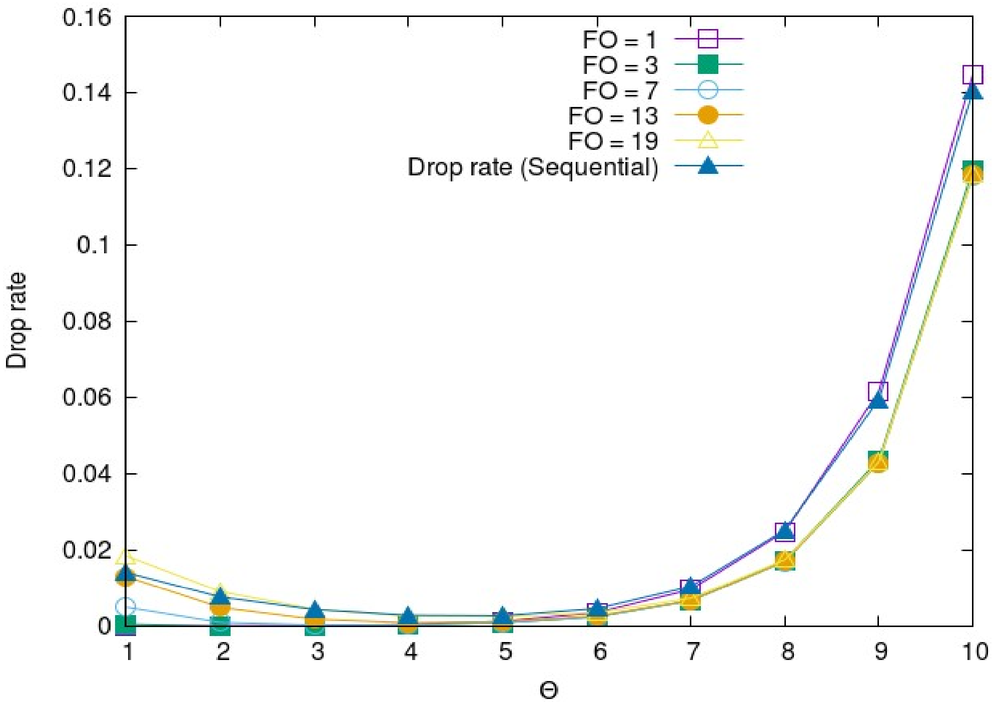
Mesh scenario

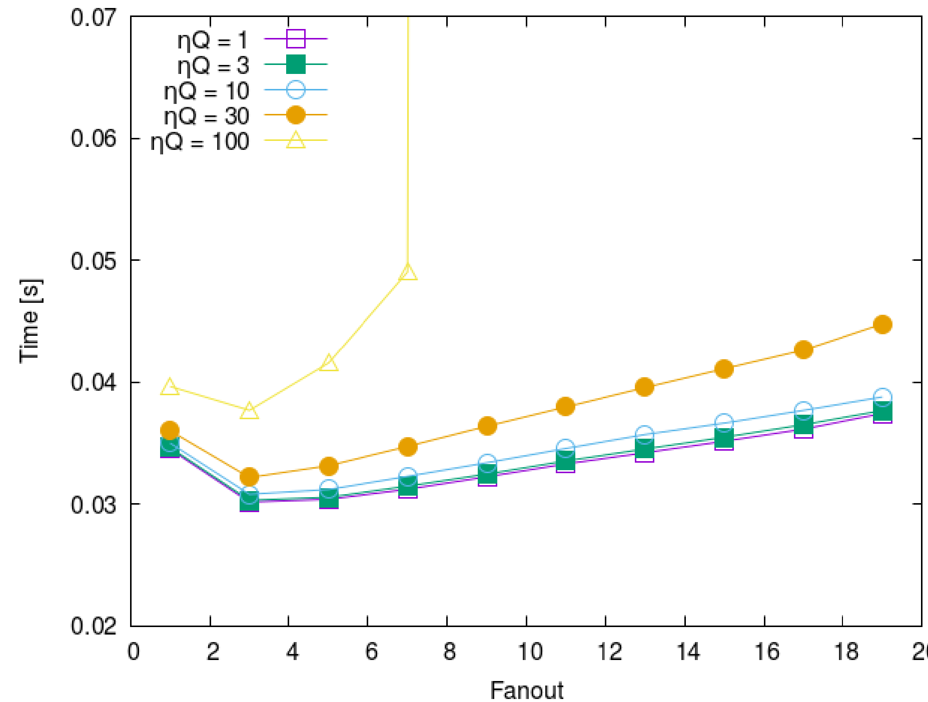




- Analysis for different Fan-Out
  - FO grows → impact of higher drop rate (apparently faster)
- Impact of η_Q (rate between probe and job size)
  - Large probe messages → high delay
  - High Fan-Out → high delay

Geographic scenario





- High variance of incoming load in fog nodes
  - Reduced impact of herding effect
  - Only few nodes start probes
- Risk of network saturation in these nodes
  - Evident when probe message is large

Conclusions

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

- Challenges of Fog computing
  - Load balancing in a distributed infrastructure
  - Impact of communication overhead
- Contributions
  - Probe-based load balancing algorithm
  - Mathematical model of delay effects
- Experimental evaluation
  - Numerical analysis on mathematical model
  - Simulation (several parameters and a realistic scenario)
- Open issues
  - Validations with prototypes
  - Proactive probing (informed protocols)

Randomized load balancing under Loosely correlated state information In fog commuting

Roberto Beraldi

“La Sapienza” University, Rome

Claudia Canali,

Riccardo Lancellotti

University of Modena and Reggio Emilia

Gabriele Proietti Mattia

“La Sapienza” University, Rome