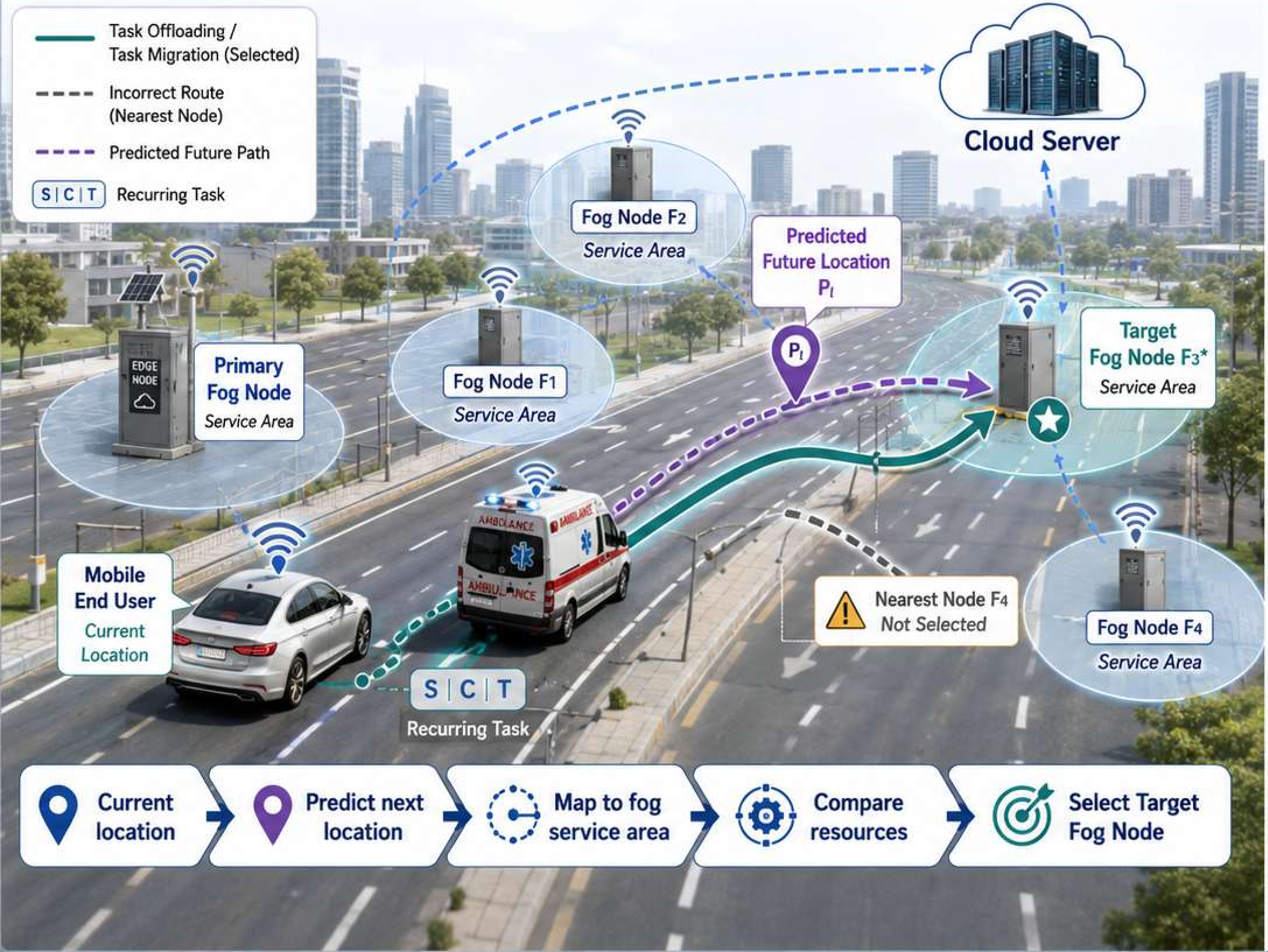
L3Fog: Predict the Location, Select the Right Fog Node

Learning-based location-aware low-latency task offloading for mobile IoT



HIGHLIGHTS OF THE PAPER

01 LINEAR-REGRESSION LOCATION PREDICTION

L3Fog predicts a mobile user's future location using time, direction, speed and coordinate data.

02 LOCATION- AND RESOURCE-AWARE SELECTION

The predicted location is mapped to nearby fog-node service areas before resource suitability is evaluated.

03 LOW-LATENCY OFFLOADING AND MIGRATION

The framework considers CPU, memory or storage, queue state and task deadline when selecting a Target Fog Node.

04 EXPERIMENTAL FINDINGS

L3Fog outperforms Lowest Latency and Lowest Distance schemes in the evaluated scenarios for migration count, handoffs, migration time and network usage. Location prediction reports 0.008% MSE.

Dataset: SUMO/LuST mobility traces | **Validation:** $p < 0.05$

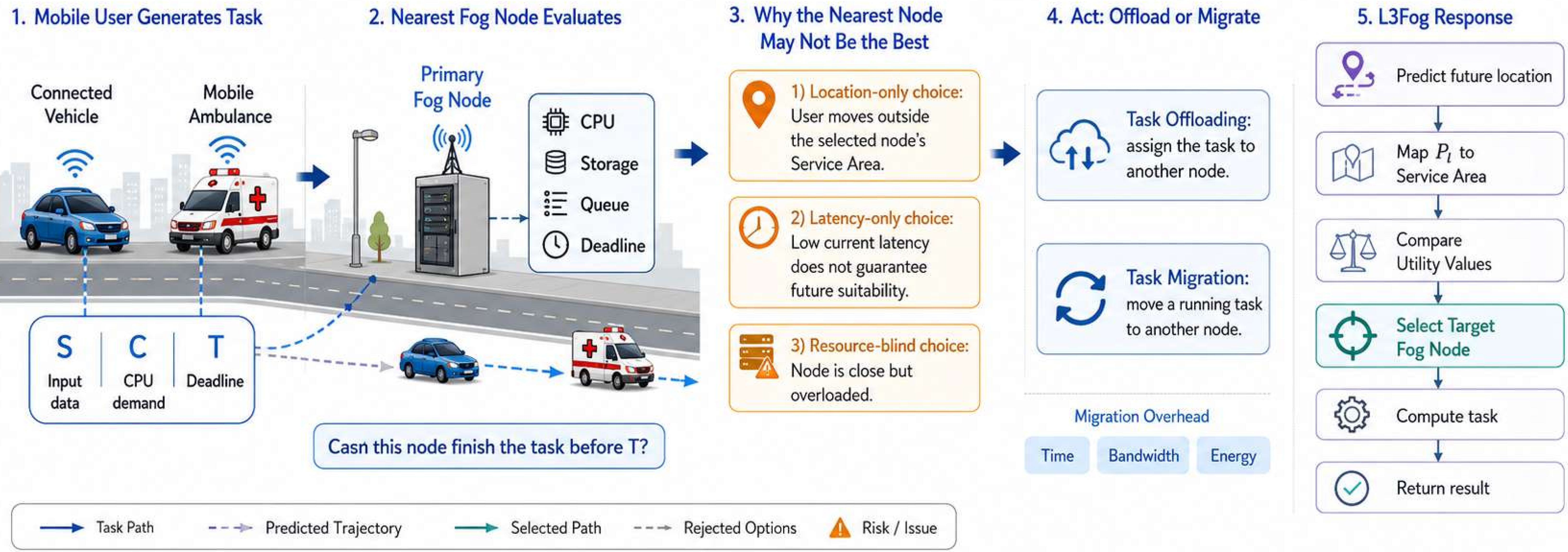
Do not choose a fog node using location alone; predict movement and verify resources.

IEEE INFOCOM Workshops 2022
DOI: 10.1109/INFOCOMWKSHPS54753.2022.9798118

Why the Nearest Fog Node Is Not Always the Best

SLIDE 2

Mobility changes location, latency and resource availability

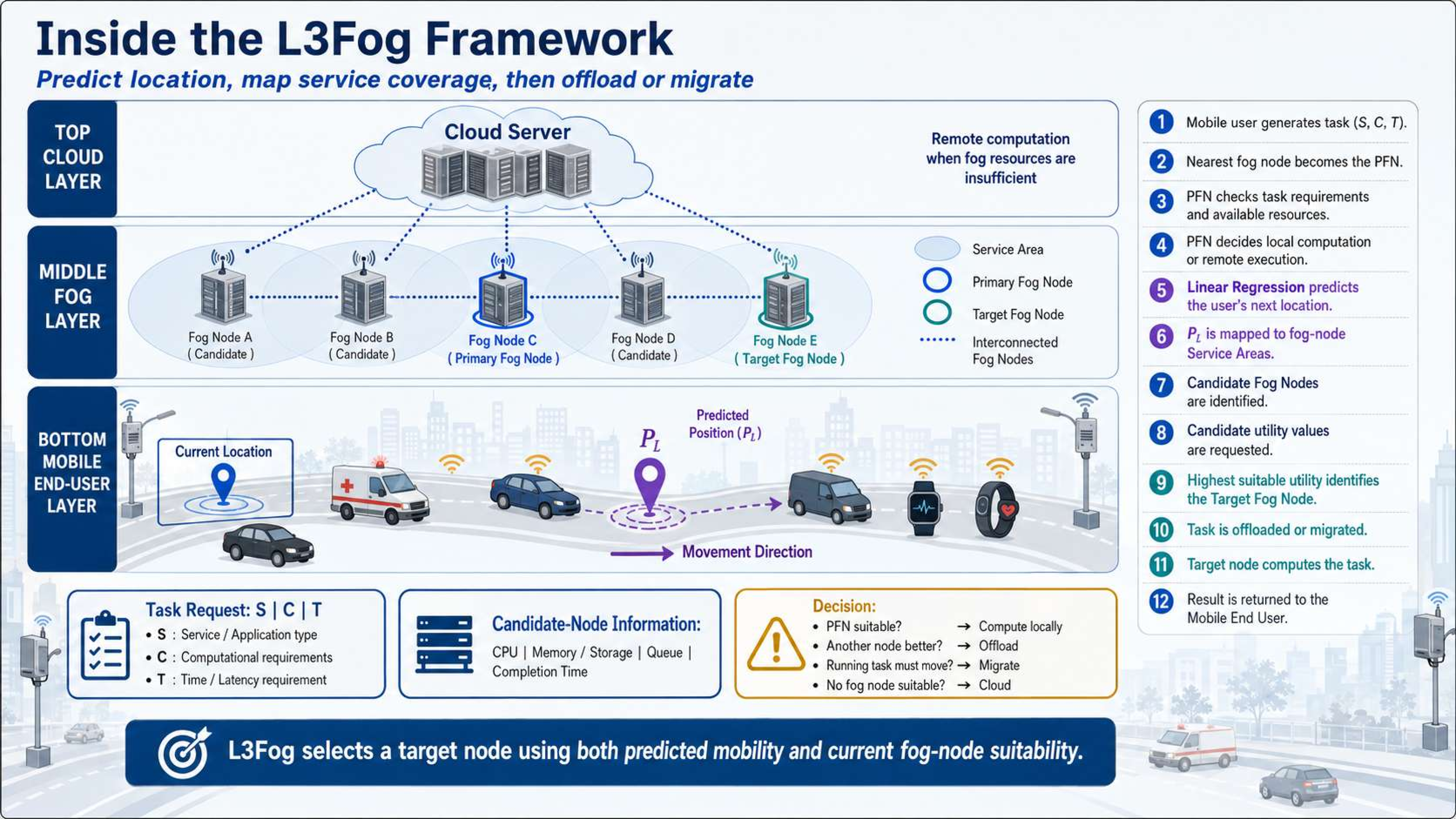


Research Gap

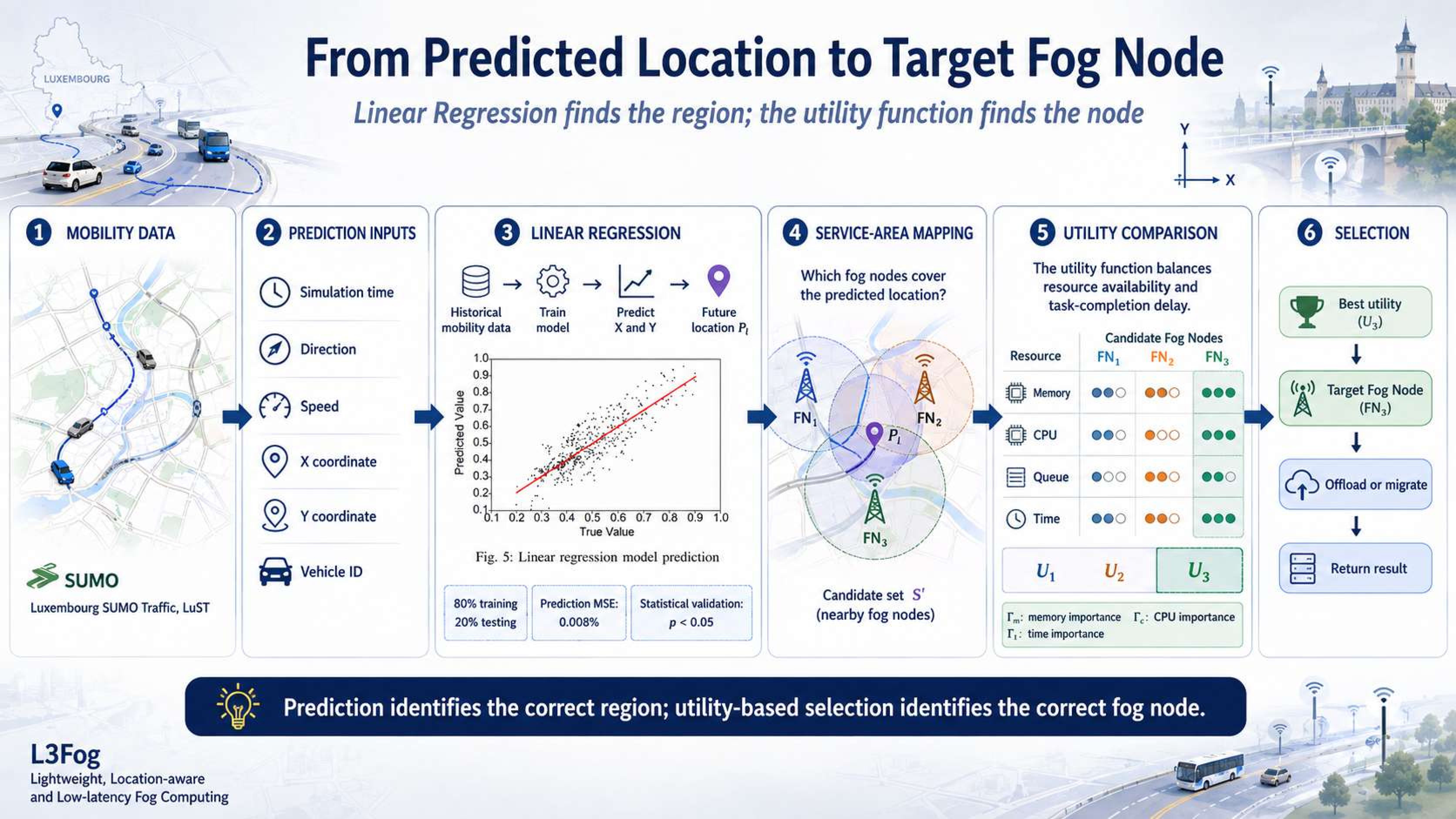
Existing approaches reduce latency, but **mapping** a **predicted** mobile-user location to an appropriate fog-node **service area** remains challenging.

Key Takeaway

L3Fog combines **future location** with **resource** and **QoS information** before selecting a node.



Inside the L3Fog Framework
Predict location, map service coverage, then offload or migrate
TOP CLOUD LAYER
Cloud Server
Remote computation when fog resources are insufficient
MIDDLE FOG LAYER
Fog Node A (Candidate)
Fog Node B (Candidate)
Fog Node C (Primary Fog Node)
Fog Node D (Candidate)
Fog Node E (Target Fog Node)
Service Area
Primary Fog Node
Target Fog Node
Interconnected Fog Nodes
BOTTOM MOBILE END-USER LAYER
Current Location
P_L
Predicted Position (P_L)
Movement Direction
1 Mobile user generates task (S, C, T).
2 Nearest fog node becomes the PFN.
3 PFN checks task requirements and available resources.
4 PFN decides local computation or remote execution.
5 Linear Regression predicts the user's next location.
6 P_L is mapped to fog-node Service Areas.
7 Candidate Fog Nodes are identified.
8 Candidate utility values are requested.
9 Highest suitable utility identifies the Target Fog Node.
10 Task is offloaded or migrated.
11 Target node computes the task.
12 Result is returned to the Mobile End User.
Task Request: S | C | T
• S : Service / Application type
• C : Computational requirements
• T : Time / Latency requirement
Candidate-Node Information:
CPU | Memory / Storage | Queue | Completion Time
Decision:
• PFN suitable? → Compute locally
• Another node better? → Offload
• Running task must move? → Migrate
• No fog node suitable? → Cloud
L3Fog selects a target node using both predicted mobility and current fog-node suitability.

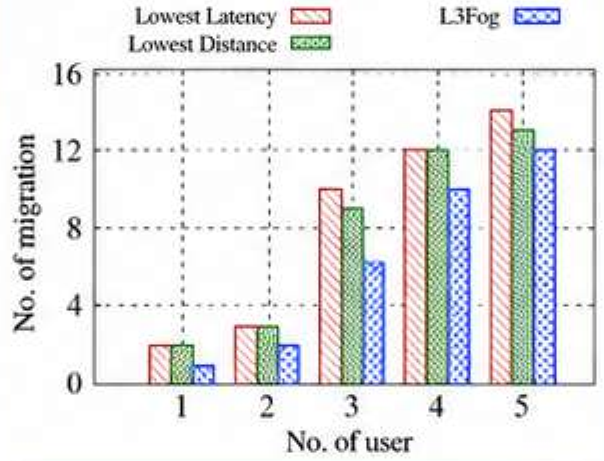


From Predicted Location to Target Fog Node
Linear Regression finds the region; the utility function finds the node
LUXEMBOURG
Y
X
1 MOBILITY DATA
SUMO
Luxembourg SUMO Traffic, LuST
2 PREDICTION INPUTS
Simulation time
Direction
Speed
X coordinate
Y coordinate
Vehicle ID
3 LINEAR REGRESSION
Historical mobility data
Train model
Predict X and Y
Future location P_l
Predicted Value
True Value
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
Fig. 5: Linear regression model prediction
80% training 20% testing
Prediction MSE: 0.008%
Statistical validation: $p < 0.05$
4 SERVICE-AREA MAPPING
Which fog nodes cover the predicted location?
FN_1
FN_2
FN_3
P_l
Candidate set S' (nearby fog nodes)
5 UTILITY COMPARISON
The utility function balances resource availability and task-completion delay.
Candidate Fog Nodes
Resource
FN_1 FN_2 FN_3
Memory
CPU
Queue
Time
U_1 U_2 U_3
Γ_m: memory importance Γ_c: CPU importance
Γ_t: time importance
6 SELECTION
Best utility (U_3)
Target Fog Node (FN_3)
Offload or migrate
Return result
Prediction identifies the correct region; utility-based selection identifies the correct fog node.
L3Fog
Lightweight, Location-aware and Low-latency Fog Computing

What the L3Fog Experiments Show

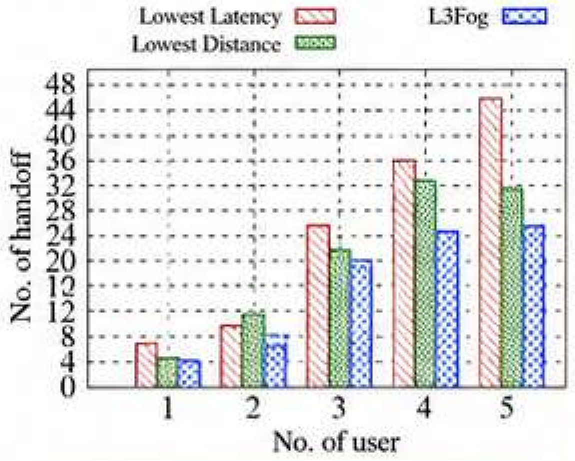
Comparison with Lowest Latency and Lowest Distance strategies

1 Number of Migrations



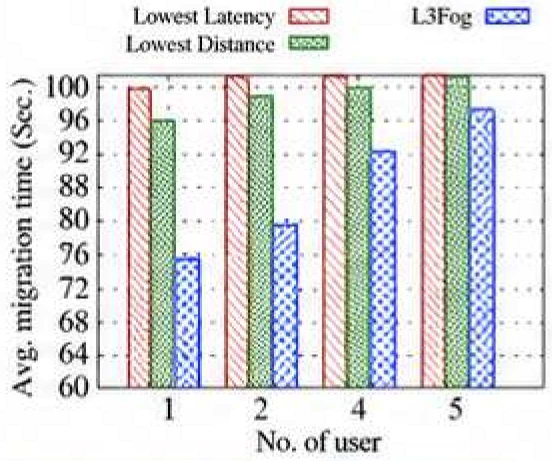
L3Fog produces fewer migrations than the baseline methods in the evaluated user scenarios.

2 Number of Handoffs



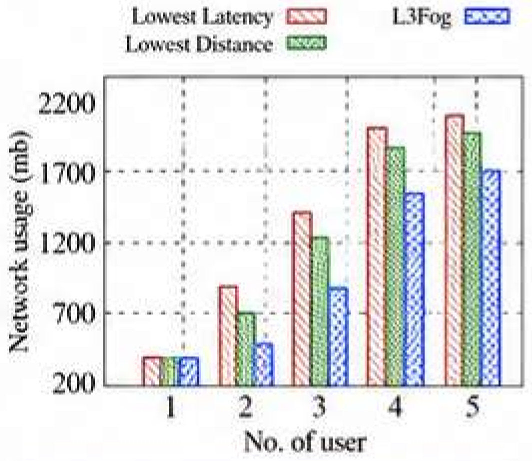
Location-aware fog selection reduces unnecessary connection changes.

3 Average Migration Time



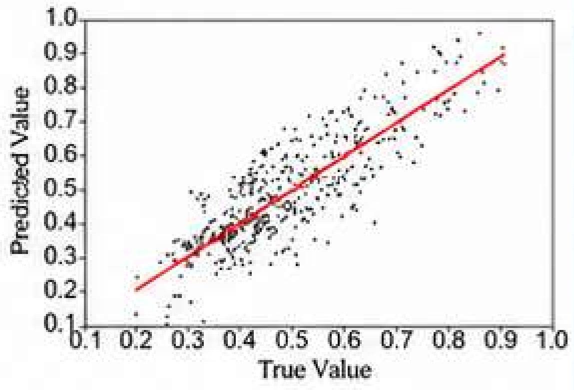
Selecting a node near the predicted location lowers migration time in the evaluated scenarios.

4 Network Usage



Improved node selection avoids repeated offloading and migration traffic.

5 Linear-Regression Prediction



0.008% MSE
$p < 0.05$

EXPERIMENTAL SETUP	MobFogSim	SUMO / LuST dataset	1 cloud node	40 fog nodes	160 mobile nodes	End-user CPU: 500 MIPS	Fog-node CPU: 2800 MIPS	Cloud CPU: 44800 MIPS	Fog-node area: 500 m × 500 m	VM size: 128 MB

BASELINE COMPARISON Compared with: Lowest Latency and Lowest Distance.

HOW L3FOG WORKS: Mobility data → Predict next location → Map service area → Compare resources and delay → Choose Target Fog Node → Reduce mobility overhead

RESEARCH CONCLUSION

L3Fog improves low-latency mobile-IoT task execution by combining location prediction with resource- and QoS-aware fog-node selection.

EVALUATION SCOPE

The results are simulation-based using MobFogSim and SUMO/LuST mobility data.

FUTURE WORK

Fault-tolerant fog-node selection | Battery-operated fog nodes | Real testbed implementation

Predict where the user is going. Select the node that can actually finish the task.