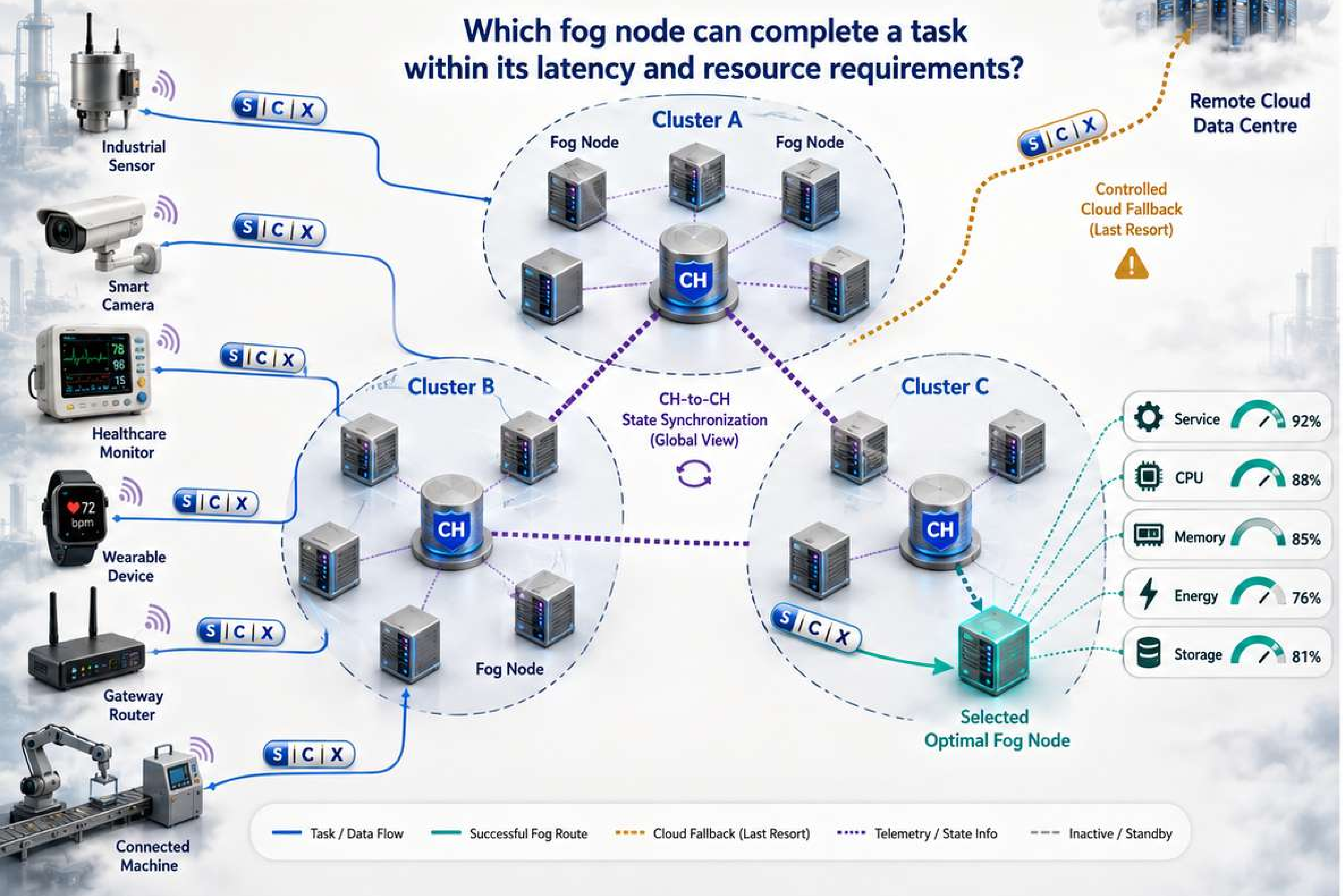
OptiFog: Finding the Right Fog Node

Global-state acquisition and resource prediction for cooperative fog networks



HIGHLIGHTS OF THE PAPER

01 GLOBAL-STATE ARCHITECTURE

Cluster Heads collect and exchange service, CPU, energy, memory and storage information across fog clusters.

02 OPTIMAL FOG-NODE SELECTION

Select a PFN or SFN that satisfies the task latency limit and has the strongest available resources.

03 CTMC-BASED PREDICTION

Predict CPU, memory and storage availability to reduce frequent state synchronization.

04 EXPERIMENTAL FINDINGS

Up to 32% lower latency and 14% lower network usage. CTMC reduces network usage by 11–24.6% and energy consumption by 7–12%.

OptiFog also sends the lowest share of tasks to the cloud compared with **DPTO** and **FogFL**.

Global awareness helps keep more computation close to IoT devices.

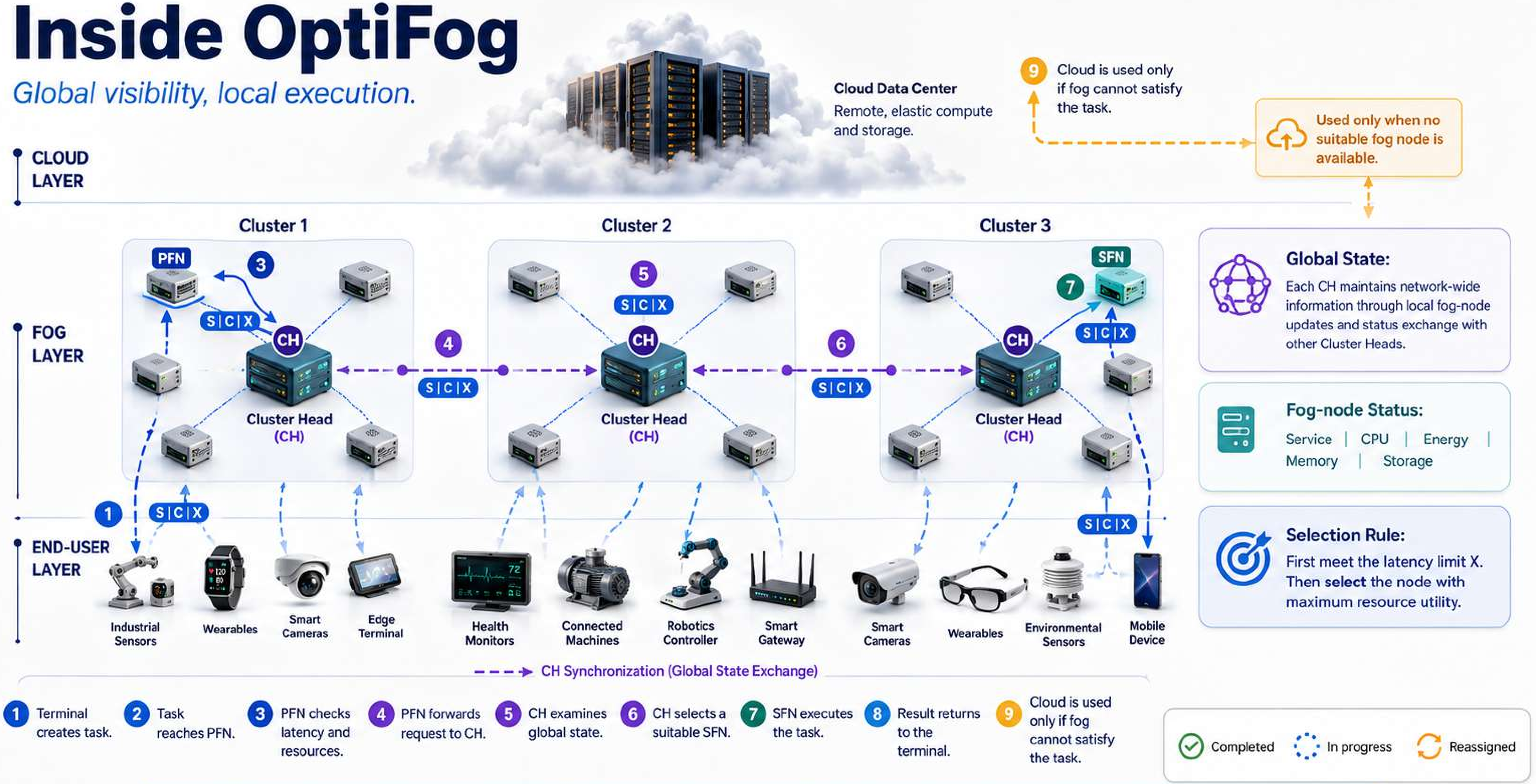
IEEE Transactions on Services Computing, 2024

DOI: 10.1109/TSC.2024.3414371

Where Should the Task Run?
Local device, Primary Fog Node, Secondary Fog Node, or cloud?
IEEE
1 Task creation at terminal
2 Local execution decision
3 Nearest Primary Fog Node (PFN)
4 Partial information (neighbour-only search)
5 OptiFog global state (global visibility via CH)
6 Execution and result return
The terminal compares local execution cost with the cost of offloading.
Execute locally
Send to nearest PFN
S C X
S: input data
C: CPU cycles
X: latency limit
PFN evaluates resources
Service availability
CPU
Memory
Energy
Storage
PFN
Can the PFN satisfy the task before X?
Search among neighbouring nodes only
Overloaded nodes
Repeated attempts
Longer communication path
Missed latency (X)
Fallback to remote cloud data centre
Existing methods often assume global state or use only neighbouring resource information.
Accepted
Processing
Completed
Result returned to terminal
PFN sends request to CH
CH observes multiple clusters
CH identifies the best SFN
Selected SFN accepts task
CH
SFN
Task stays in the fog layer
Global resource visibility allows direct selection of a suitable SFN.
72
98%
6,243
Industrial Automation
OptiFog searches beyond neighbouring nodes while still meeting task QoS requirements.
Healthcare & Wearables

Inside OptiFog

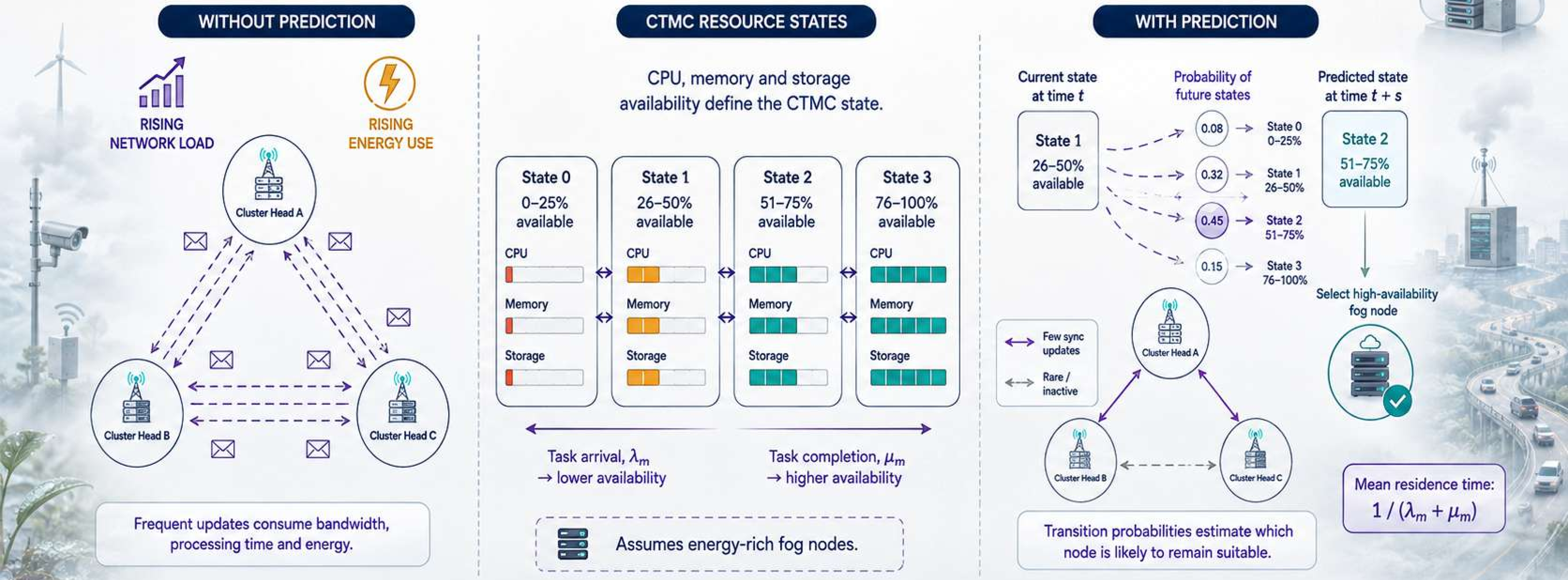
Global visibility, local execution.



OptiFog combines state acquisition, node selection, task tracking, failure handling and load balancing.

Predict Availability Instead of Checking Constantly

CTMC estimates future resource states and reduces synchronization overhead



Instead of repeatedly asking every node, estimate which node is likely to remain available.

CTMC maintains useful state awareness with fewer status exchanges.

What the Experiments Show

Performance comparison of OptiFog, DPTO and FogFL

Global state → better SFN selection → more fog execution → lower latency and traffic → lower cloud dependence

Latency (task execution)

Fig. 3: Latency vs. number of tasks

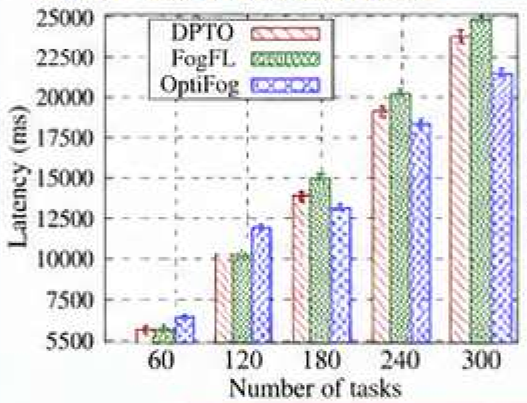
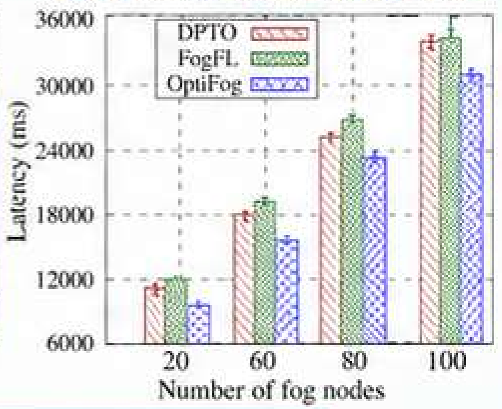


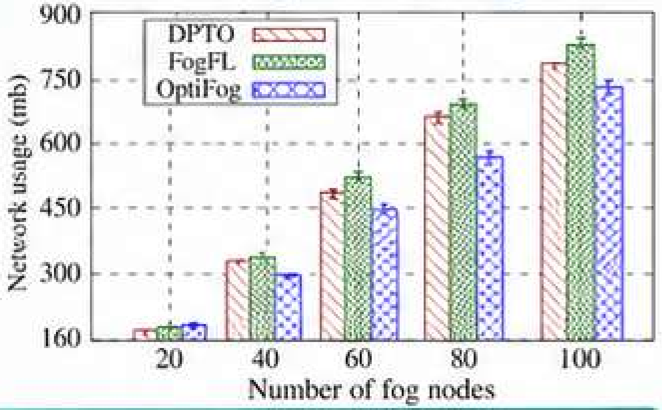
Fig. 4: Latency vs. number of fog nodes



Up to 32% lower latency

Network Usage (traffic)

Fig. 5: Network usage vs. number of fog nodes



Up to 14% lower network usage

Cloud Offloading (tasks sent to cloud)

Fig. 6(a): % tasks offloaded vs. # tasks

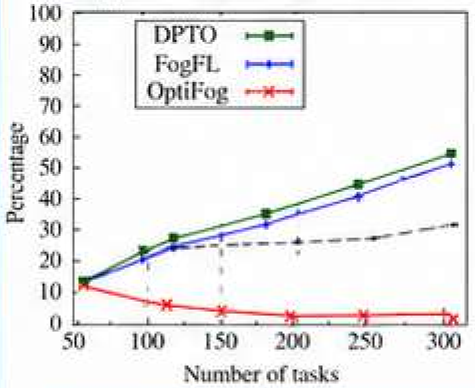
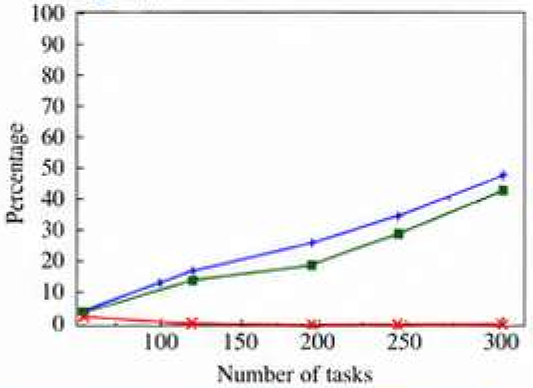


Fig. 6(b): % tasks offloaded vs. # tasks



OptiFog sends the lowest share of tasks to the cloud

CTMC Impact (OptiFog with vs. without CTMC)

Fig. 7: Network usage vs. number of tasks

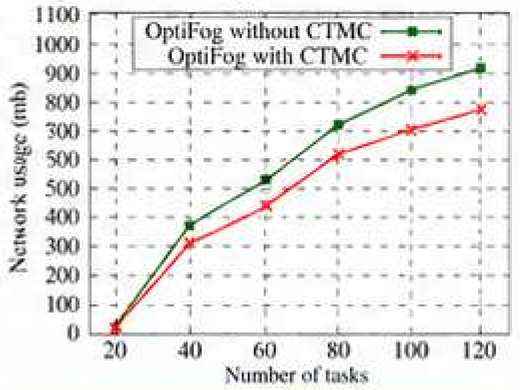
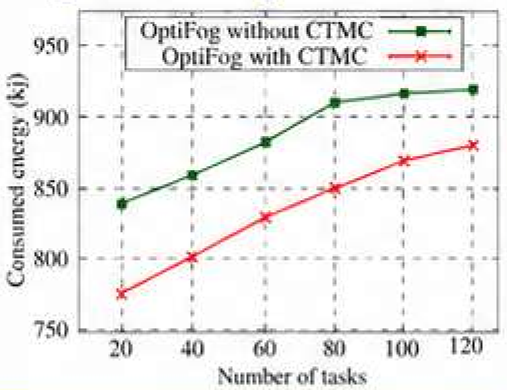


Fig. 8: Consumed energy vs. number of tasks

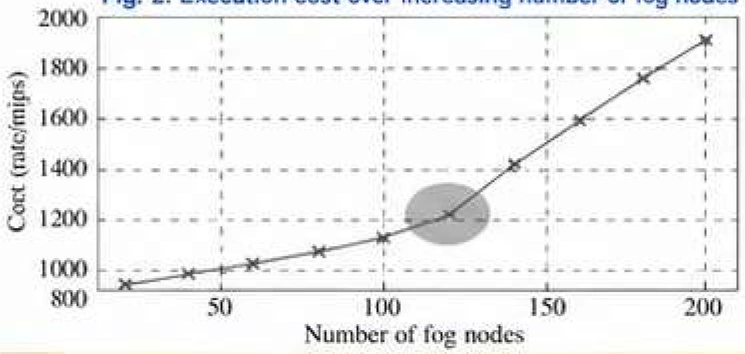


11–24.6% lower network usage

7–12% lower energy consumption

Scalability Limitation (global-state acquisition cost)

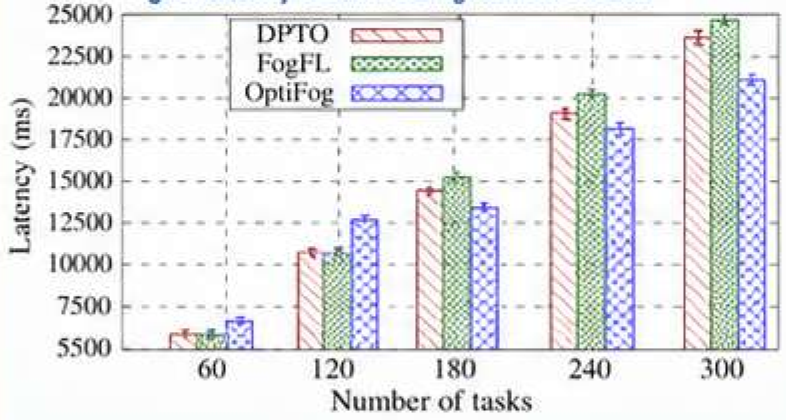
Fig. 2: Execution cost over increasing number of fog nodes



Global-state acquisition cost rises sharply as the fog network grows. Experiments used up to 120 fog nodes. This is an experimental design choice, not a universal hard limit.

Baseline Scalability (latency vs. tasks)

Fig. 3: Latency over increasing number of tasks



Experimental Setup: *iFogSim | 5–120 fog nodes | 60–300 tasks | 420 KB task size | 500 m × 500 m deployment | 100 simulation runs | comparison: DPTO and DPTO and FogFL*

Final Conclusion: *OptiFog improves load balancing and resource utilisation by selecting a suitable fog node using network-wide resource awareness.*

Future Work: *Machine-learning-based resource-availability prediction.*

Key Takeaway: ***Know the network. Predict availability. Keep more computation in the fog.***