



Performance Evaluation of Edge Computing Architectures for Industrial Automation

Galadriel Mardaniyan

Department of Electrical and Computer Engineering, Kavir Polytechnic University of Technology, Iran
galadriel.mardaniyan@kput-ir.net

Peer Review Information	Abstract
<i>Submission: 20 March 2018</i> <i>Revision: 06 April 2018</i> <i>Acceptance: 26 April 2018</i> Keywords <i>Edge Computing, Industrial Automation, Industrial Internet of Things, Fog Computing, Performance Evaluation</i>	Industrial automation increasingly connects sensors, programmable logic controllers (PLCs), robots, vision systems, and enterprise services. A cloud-only architecture provides elastic computation but can introduce network delay, jitter, bandwidth consumption, and dependence on wide-area connectivity. Edge and fog computing move selected processing closer to machines and therefore offer a promising architecture for latency-sensitive industrial applications. This paper presents a simulation-based methodology for evaluating three deployment alternatives: cloud-only processing, a single-edge gateway, and a hierarchical edge-fog-cloud architecture. The study uses an identical workload of 30,000 industrial request-response transactions and evaluates mean end-to-end latency, 95th-percentile latency, jitter, success under a 50 ms control deadline, transaction availability, and the proportion of traffic forwarded over the wide-area network (WAN). The simulation indicates that the single-edge architecture reduces mean latency from 82.98 ms to 15.85 ms, while the hierarchical architecture reduces it to 10.43 ms. WAN traffic is reduced to 34% and 18% of the cloud-only baseline, respectively. The findings support the use of localized processing for time-sensitive automation while retaining cloud services for long-term analytics, model management, and enterprise integration. Because the evaluation is simulation-based, the numerical results should be interpreted as reproducible comparative evidence under the stated assumptions rather than as measurements from a particular factory.

Introduction

Industrial automation systems are evolving from isolated control networks toward highly connected cyber-physical environments in which field devices, controllers, industrial computers, and enterprise applications exchange large volumes of operational data. The Industrial Internet of Things (IIoT) extends sensing, identification, networking, and computation into industrial processes, enabling condition monitoring, predictive maintenance, quality control, production optimization, and remote supervision [2]. However, the performance requirements of industrial control differ from

conventional information technology. Many automation tasks require bounded latency, low jitter, high availability, and predictable behavior even when connectivity to remote services is degraded.

Traditional cloud computing centralizes computation and storage in remote data centers. This arrangement is effective for large-scale analytics and elastic resource provisioning, but it may be unsuitable for short control loops because every request must cross local and wide-area networks before a response is returned. Fog computing was introduced as an extension of

cloud capabilities toward the network edge, emphasizing low latency, geographic distribution, and support for real-time sensor-actuator applications [1]. Later work formalized edge computing as a paradigm in which data are processed near their sources to improve response time, reduce bandwidth consumption, and strengthen privacy [4]. Fog and edge concepts therefore provide architectural alternatives for industrial environments in which local responsiveness and continued operation are essential.

The industrial context also requires deterministic communication. Wollschlaeger et al. [8] described the transition of industrial communication toward the Internet of Things and Industry 4.0, highlighting the need to integrate established automation networks with Internet technologies without losing real-time properties. Pop et al. [11] later argued that Time-Sensitive Networking (TSN) can serve as deterministic transport for fog computing in industrial automation. In parallel, O'Donovan et al. [10] demonstrated the suitability of fog-based industrial cyber-physical systems for embedded low-latency analytics, and Aazam et al. [12] discussed fog as middleware for local processing within IIoT and Industry 4.0 environments.

Accordingly, the objective of this paper is to define and apply a clear performance-evaluation methodology that compares three computing placements under a common industrial workload. The study asks: (1) how much can edge placement reduce end-to-end latency and jitter; (2) how does architecture placement affect completion within a 50 ms deadline; (3) how much WAN traffic can be avoided through local filtering and decision making; and (4) what trade-off is created between architectural simplicity and hierarchical coordination? The contribution is methodological: it specifies the architecture models, workload, metrics, simulation assumptions, evaluation steps, and interpretation rules required for a reproducible comparison using literature available up to 2018.

Literature Review

The evolution from cloud-centric computing to edge-oriented architectures is well established in the pre-2019 literature. Bonomi et al. [1] introduced fog computing as a distributed layer

between devices and cloud services, identifying latency, location awareness, mobility, and real-time streaming as defining characteristics. Xu et al. [2] surveyed IoT applications in industrial environments and showed how sensing, wireless communication, and cloud technologies were becoming integrated with industrial systems. Yi et al. [3] provided an early survey of fog concepts, applications, and implementation issues, while Shi et al. [4] emphasized response time, bandwidth savings, and data safety as central motivations for edge computing.

Chiang and Zhang [5] positioned fog along the cloud-to-things continuum and discussed distributed control, networking, storage, and computation for IoT. Satyanarayanan [6] explained why proximity matters, particularly for highly responsive services, scalability, privacy-policy enforcement, and resilience to cloud outages. Mao et al. [7] reviewed mobile edge computing from the communication perspective and highlighted computation offloading and joint communication-computing resource management. Although mobile-edge work is not specific to factories, its analytical treatment of placement and offloading is directly relevant to industrial workload distribution.

Industrial publications sharpened these concepts. Wollschlaeger et al. [8] reviewed automation networking in the context of Industry 4.0. Boyes et al. [9] described IIoT as an architecture involving networked smart objects, cyber-physical assets, and optional cloud or edge platforms, while also emphasizing that device location and connectivity matter to system analysis. O'Donovan et al. [10] presented a fog-based industrial cyber-physical system for low-latency machine-learning applications and reported substantially lower worst-case execution latency for the local fog interface than for a cloud interface under stress. Pop et al. [11] connected fog computing to deterministic TSN transport for hard real-time industrial communication. Finally, Aazam et al. [12] argued that fog can locally process urgent industrial tasks and filter data before transmission to cloud services. Together, these works motivate an evaluation framework that treats computing placement, network path, and workload localization as measurable architectural variables.

Table 1. Literature foundation for the proposed performance evaluation.

Study	Year	Architecture focus	Relevance to evaluation
Bonomi et al. [1]	2012	Fog layer near devices	Low latency and real-time IoT
Xu et al. [2]	2014	Industrial IoT	Industrial sensing and connectivity

Yi et al. [3]	2015	Fog survey	Concepts, applications, implementation issues
Shi et al. [4]	2016	Edge computing	Response time and bandwidth reduction
Chiang & Zhang [5]	2016	Cloud-to-things fog	Distributed compute/control/storage
Satyanarayanan [6]	2017	Proximate edge	Responsiveness and outage resilience
Mao et al. [7]	2017	MEC/offloading	Communication-computation placement
Wollschlaeger et al. [8]	2017	Industrial networks	Industry 4.0 communication requirements
Boyes et al. [9]	2018	IIoT framework	Device location and connectivity
O'Donovan et al. [10]	2018	Industrial fog CPS	Low-latency embedded analytics
Pop et al. [11]	2018	Fog + TSN	Deterministic industrial transport
Aazam et al. [12]	2018	Fog for IIoT	Local processing and data reduction

Methodology

1. Research Design and Architecture Models

The study adopts a controlled comparative simulation. Each architecture receives exactly the same number of transactions and is assessed with the same metric definitions. Only the placement of computation, the corresponding network path, processing delay distribution, queueing behavior, and WAN forwarding ratio change between alternatives. This isolates the architectural effect and avoids comparing unrelated hardware platforms or workloads. Three models are evaluated:

- Cloud-only architecture: field events are transmitted to a remote cloud service for

processing and the response is returned through the WAN. This model represents maximum centralization and uses 100% WAN forwarding.

- Single-edge gateway architecture: time-sensitive processing is executed at one plant-local edge gateway; summarized or selected data are sent to cloud services. The WAN forwarding share is modeled as 34%.
- Hierarchical edge-fog-cloud architecture: device/cell edge nodes perform immediate processing, a plant-level fog tier coordinates cross-cell services, and the cloud performs long-term analytics and model management. The WAN forwarding share is modeled as 18%.

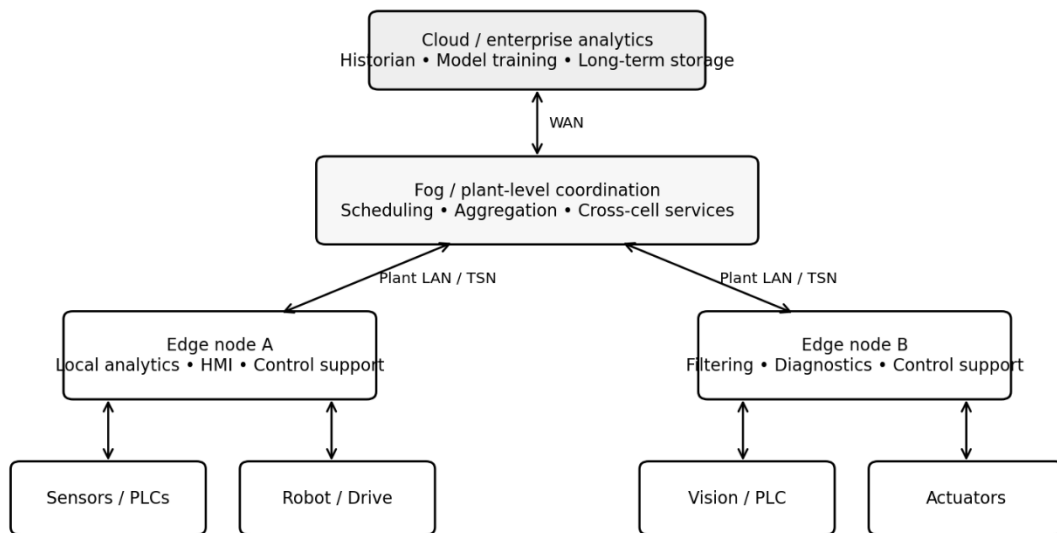


Figure 1. Layered edge-fog-cloud architecture used as the reference model for industrial automation.

2. Workload and Simulation Assumptions

A synthetic industrial request-response workload of 30,000 transactions is generated with a fixed random seed of 2018 to support reproducibility. A transaction represents a sensor/PLC event that requires computation and

a response used by an HMI, controller, actuator, diagnostic service, or supervisory application. The evaluation focuses on communication-and-processing latency rather than detailed plant physics. Completed-transaction latency is modeled as the sum of uplink/network delay, processing delay, downlink delay, and occasional

queueing delay. Component delays are sampled from truncated normal distributions so that non-physical negative delays cannot occur. Queueing bursts are introduced probabilistically to represent transient congestion and shared-resource contention.

The cloud-only model uses the longest network path and the largest queueing bursts. The single-edge gateway shortens the communication path but concentrates local requests on one gateway. The hierarchical model uses the shortest device-

to-edge path and distributes processing across local and plant-level resources. A small architecture-specific transaction-failure probability represents temporary service/network unavailability. Failed transactions contribute to the availability metric but are excluded from completed-transaction latency statistics. These assumptions are not intended to reproduce any specific vendor system; instead, they instantiate the proximity and decentralization effects consistently described in [1], [4]-[7], [10], and [12].

Table 2. Simulation parameters. Values are mean $\pm$ standard deviation for truncated normal delay components.

Architecture	Uplink ms	Process ms	Downlink ms	Queue burst	Failure	WAN share
Cloud-only	36 $\pm$ 9	11 $\pm$ 3	34 $\pm$ 8	8%	1.8%	100%
Single-edge gateway	4.8 $\pm$ 1.4	6.5 $\pm$ 2.0	4.2 $\pm$ 1.2	5%	0.6%	34%
Hierarchical edge-fog-cloud	2.8 $\pm$ 0.9	5.0 $\pm$ 1.5	2.5 $\pm$ 0.8	3.5%	0.3%	18%

3. Performance Metrics and Step-by-Step Procedure

Six metrics are used. Mean end-to-end latency measures typical response time; the 95th percentile captures tail behavior; jitter is represented by the sample standard deviation of completed latency; deadline success is the proportion of all generated transactions that both complete successfully and finish within 50 ms; transaction availability is the successful-completion proportion irrespective of deadline; and WAN traffic share is the percentage of operational traffic forwarded beyond the plant network. These metrics combine responsiveness, determinism, reliability, and bandwidth efficiency, which are central concerns in industrial edge literature [8], [10], [11].

1. Initialize the random generator with seed 2018 and create 30,000 identical logical transactions for each architecture.
2. Sample uplink, processing, and downlink delays from the architecture-specific distributions shown in Table 2.
3. Inject transient queueing events using the stated queue-burst probability and architecture-specific gamma-distributed extra delay.
4. Sample transaction failures independently according to the architecture-specific failure probability.

5. For successful transactions, calculate end-to-end latency and summarize the mean, 95th percentile, and standard deviation.
6. For all generated transactions, calculate success within the 50 ms deadline and overall transaction availability.
7. Apply the predefined WAN-forwarding ratios and compare each edge architecture with the cloud-only baseline.
8. Interpret improvements together with architectural trade-offs rather than selecting an architecture using latency alone.

Results

Table 3 reports the simulation output. The cloud-only architecture produced a mean completed-transaction latency of 82.98 ms and a 95th-percentile latency of 107.17 ms. Only 0.58% of all generated transactions completed successfully inside the 50 ms deadline, although overall transaction availability remained 98.20%. The result illustrates an important distinction: a transaction can eventually succeed but still be too late for a strict control deadline. This is consistent with the motivation for proximate processing described by Shi et al. [4] and Satyanarayanan [6].

Table 3. Simulated performance results for 30,000 transactions per architecture.

Architecture	Mean ms	P95 ms	Jitter ms	<50 ms %	Avail. %	WAN %
Cloud-only	82.98	107.17	14.74	0.58	98.20	100
Single-edge gateway	15.85	20.95	3.32	99.36	99.36	34

Hierarchical edge-fog-cloud	10.43	13.80	2.12	99.71	99.71	18
-----------------------------	-------	-------	------	-------	-------	----

Moving time-sensitive computation to a single-edge gateway reduced mean latency by 80.90% relative to cloud-only processing, from 82.98 ms to 15.85 ms. The 95th percentile fell by 80.45% and jitter fell by 77.48%. Deadline success increased to 99.36%, and only 34% of traffic was modeled as requiring WAN forwarding. These gains reflect the removal of the remote round trip from the critical response path and correspond to the local-processing rationale advanced in fog and edge research [1], [5], [12].

The hierarchical edge-fog-cloud model achieved the strongest simulated performance: mean

latency of 10.43 ms, 95th percentile of 13.80 ms, and jitter of 2.12 ms. Relative to cloud-only processing, this represents reductions of 87.43%, 87.13%, and 85.63%, respectively. Deadline success reached 99.71%, transaction availability reached 99.71%, and the WAN traffic share was only 18%. The hierarchy therefore combines local immediacy with a coordination tier that can aggregate plant data before cloud transfer. This architectural separation is compatible with industrial designs that retain cloud functions while moving urgent processing closer to machines [9], [10], [12].

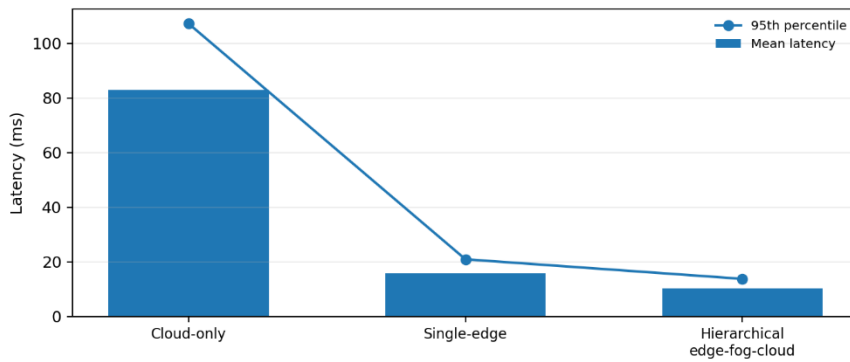


Figure 2. Mean and 95th-percentile latency from the reproducible simulation benchmark.

The results should not be interpreted as proof that every hierarchical deployment will outperform every single-edge implementation. Local nodes may be resource constrained, and hierarchy introduces additional orchestration, configuration, security, and lifecycle-management requirements. Deterministic networking is also essential when computation placement alone cannot guarantee bounded delivery time; this is why TSN-based approaches remain significant for industrial fog architectures [11]. Furthermore, cloud resources remain valuable for historical data storage, fleet-wide optimization, model training, and computationally intensive analysis. Therefore, the practical goal is not to eliminate cloud computing but to place each function at the tier whose latency, reliability, privacy, and resource characteristics match the industrial requirement.

Conclusion and Discussion

This methodology paper evaluated cloud-only, single-edge, and hierarchical edge-fog-cloud architectures for industrial automation using a controlled simulation of 30,000 transactions per alternative. Under the stated assumptions, edge placement substantially improved

responsiveness and reduced WAN dependence. The single-edge model reduced mean latency by about 80.9%, while the hierarchical architecture reduced it by about 87.4% and achieved the highest deadline success. These outcomes reinforce the central pre-2019 literature argument that proximity and localized processing are valuable when industrial applications have strict response-time and bandwidth constraints [4]-[6]. From an engineering perspective, the hierarchical model is most suitable when a plant contains several cells or production areas that require both independent low-latency decisions and shared coordination. A single-edge gateway is simpler and may be sufficient for smaller installations, but it can become a resource bottleneck or single concentration point. Cloud-only processing remains appropriate for non-real-time functions, large-scale optimization, archiving, enterprise integration, and model training. A balanced industrial architecture therefore assigns fast control support, filtering, diagnostics, and local inference to edge/fog tiers while reserving the cloud for tasks whose value exceeds their latency sensitivity. The principal limitation is that this study uses a synthetic latency and failure model rather than a physical factory testbed. Real

performance depends on PLC scan cycles, protocol stacks, TSN configuration, virtualization overhead, processor type, workload interference, message size, topology, wireless conditions, and security mechanisms. The numerical values in Table 3 are therefore comparative simulation results, not field measurements. Future validation should reproduce the same methodology using industrial PCs or gateways, OPC UA and/or MQTT traffic, hardware-in-the-loop controllers, and TSN-capable Ethernet. Additional experiments should vary workload intensity, failure duration, number of devices, task placement, security overhead, and energy consumption. Even with these limitations, the methodology provides a transparent and reproducible basis for selecting an edge computing architecture before costly plant deployment.

References

- F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, 2012, pp. 13–16. DOI: 10.1145/2342509.2342513.
- L. D. Xu, W. He, and S. Li, "Internet of Things in Industries: A Survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014. DOI: 10.1109/TII.2014.2300753.
- S. Yi, C. Li, and Q. Li, "A Survey of Fog Computing: Concepts, Applications and Issues," in *Proceedings of the 2015 Workshop on Mobile Big Data*, 2015, pp. 37–42. DOI: 10.1145/2757384.2757397.
- W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016. DOI: 10.1109/JIOT.2016.2579198.
- M. Chiang and T. Zhang, "Fog and IoT: An Overview of Research Opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016. DOI: 10.1109/JIOT.2016.2584538.
- M. Satyanarayanan, "The Emergence of Edge Computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017. DOI: 10.1109/MC.2017.9.
- Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017. DOI: 10.1109/COMST.2017.2745201.
- M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017. DOI: 10.1109/MIE.2017.2649104.
- H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The Industrial Internet of Things (IIoT): An Analysis Framework," *Computers in Industry*, vol. 101, pp. 1–12, 2018. DOI: 10.1016/j.compind.2018.04.015.
- P. O'Donovan, C. Gallagher, K. Bruton, and D. T. J. O'Sullivan, "A Fog Computing Industrial Cyber-Physical System for Embedded Low-Latency Machine Learning Industry 4.0 Applications," *Manufacturing Letters*, vol. 15, pp. 139–142, 2018. DOI: 10.1016/j.mfglet.2018.01.005.
- P. Pop, M. L. Raagaard, M. Gutierrez, and W. Steiner, "Enabling Fog Computing for Industrial Automation Through Time-Sensitive Networking (TSN)," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 55–61, 2018. DOI: 10.1109/MCOMSTD.2018.1700057.
- M. Aazam, S. Zeadally, and K. A. Harras, "Deploying Fog Computing in Industrial Internet of Things and Industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4674–4682, 2018. DOI: 10.1109/TII.2018.2855198.