Edge Computing Benefits in Low-Latency IoT Applications: A Comprehensive Survey

Prof. Claudio Enrico Palazzi   
*Department of Mathematics*

*University of Padua*

*Padua, Italy*  
cpalazzi@math.unipd.it

Michael Amista’  
*Department of Mathematics*

*University of Padua*

*Padua, Italy*  
michael.amista@studenti.unipd.it

*Abstract*—The rapid proliferation of Internet of Things (IoT) devices has generated unprecedented volumes of data, creating significant challenges in computational processing, latency, and resource management. This comprehensive survey explores the transformative role of edge computing in addressing critical limitations of traditional cloud-based IoT architectures. By examining recent advances in edge computing technologies, this work provides insights on how computational capabilities can be extended from centralized cloud infrastructure to the network's edge, enabling more efficient, responsive, and intelligent IoT applications.

The investigation reveals that edge computing fundamentally reshapes IoT capabilities by bringing computational resources closer to data sources, thereby mitigating network bandwidth constraints and communication latency. The survey systematically analyses edge computing architectures across diverse domains, including mobile applications, industrial manufacturing, and interactive technologies. Particular emphasis is placed on mobile edge computing (MEC) paradigm that addresses energy efficiency, computation offloading, and low-latency requirements. Key findings demonstrate edge computing's potential to support real-time processing, enhance computational autonomy, and optimize resource utilization for resource-constrained smart devices.

Furthermore, the survey identifies persistent research challenges and opportunities, highlighting the need for continued innovation in edge computing infrastructure, communication protocols, and intelligent resource management strategies. By providing a comprehensive overview of current technological landscapes, this study offers researchers and practitioners a critical reference for understanding and implementing edge computing solutions in IoT ecosystems.

Keywords—Edge computing, Internet of Things (IoT), Low-Latency Applications, Mobile game, IoT-Based Manufacturing, Mobile Edge Computing (MEC), Real-Time Processing, Energy Efficiency

# Introduction: context and case studies

The Internet of Things (IoT) represents a transformative technological paradigm that has rapidly evolved over the past decade. At its core, IoT encompasses a vast network of interconnected devices equipped with sensors, software, and network connectivity, enabling them to collect, exchange, and process data. From smart home devices and wearable technologies to industrial sensors and urban infrastructure monitoring systems, IoT has emerged as a critical technological ecosystem that bridges the physical and digital worlds.

The proliferation of IoT devices has been exponential, with global estimates suggesting over 75 billion connected devices by 2025. These devices range from simple sensors measuring environmental parameters to complex systems managing critical infrastructure. However, the initial generation of IoT devices was predominantly limited to data collection and transmission, with minimal on-site computational capabilities.

## Limitations of traditional cloud computing

Traditional cloud computing models have been the primary approach for processing IoT-generated data. In this centralized paradigm, devices collect data and transmit it to remote data centres for processing and analysis. While this approach worked effectively for early IoT applications, it has become increasingly inadequate for emerging low-latency use cases.

The primary limitations of traditional cloud computing in IoT contexts include:

* **Network Bandwidth Constraints**: Transmitting large volumes of data from numerous devices to centralized cloud servers creates significant network congestion and bandwidth challenges.
* **Communication Latency**: The physical distance between IoT devices and cloud data centres introduces substantial processing delays, rendering the approach unsuitable for low-latency applications.
* **Resource Inefficiency**: Sending all collected data to remote servers for processing is computationally and energetically inefficient, especially for resource-constrained devices.
* **Privacy and Security Concerns**: Continuous data transmission to external servers raises critical questions about data privacy and potential security vulnerabilities.

## Emergence of edge computing

Edge computing has emerged as a revolutionary solution to address the inherent limitations of traditional cloud-based IoT architectures. By bringing computational capabilities closer to the data sources, directly at the network's edge, this paradigm fundamentally transforms how IoT systems process and analyse information.

Key characteristics of edge computing include:

* Distributed computational resources
* Local data processing and filtering
* Reduced latency
* Enhanced real-time decision-making capabilities
* Improved energy efficiency
* Increased system autonomy

The technological advancements in embedded systems have been fundamental in enabling edge computing. Modern IoT devices now possess sufficient computational resources to perform complex data processing tasks locally, marking a significant departure from earlier generations of limited-capability sensors.

## Paper scope and contribution

This comprehensive survey provides a systematic exploration of edge computing technologies within the IoT ecosystem, addressing critical computational challenges in modern low-latency IoT applications.

The survey is structured around three primary research dimensions. First, it examines the architectural transformation from centralized cloud computing to distributed edge computing architectures. This exploration investigates technological advancements that enable more efficient, responsive, and intelligent IoT systems by bringing computational capabilities closer to data sources.

Second, the research systematically explores edge computing implementations across multiple technological domains. These include mobile applications, industrial manufacturing, interactive technologies, and mobile edge computing frameworks. By analysing these diverse contexts, the survey provides a holistic understanding of edge computing's versatility and potential.

Third, the survey emphasizes performance optimization by highlighting how edge computing addresses fundamental limitations in IoT systems. This includes strategies for mitigating network bandwidth constraints, reducing communication latency, supporting real-time processing, enhancing computational autonomy, and optimizing resource utilization for resource-constrained devices.

As last point, this work provides insights on current research challenges and opportunities to promote the continuous investigation and innovation in edge computing infrastructures.

# Edge computing in internet of things

This section provides a complete overview on edge computing, highlighting the main features, its role in IoT networks, key requirements for successful deployment and the different computing paradigms. This section also provides some edge computing applications that witness the strength of this paradigm in IoT low-latency applications.

## Role of edge computing

Edge computing emerges as a strategic technological paradigm that fundamentally transforms how Internet of Things (IoT) systems process and interact with data. As highlighted by Hassan et al. [1], edge computing plays multiple critical roles in IoT ecosystems:

**Data Acquisition and Processing**: Edge devices, including sensors and intelligent machines, now possess the capability to capture streaming data and perform immediate analysis. This approach aligns with the evolving computational philosophy of "moving the algorithm to the data" rather than transporting data to centralized algorithms. For instance, in smart transportation systems, traffic light cameras can simultaneously capture and analyse data, enabling instantaneous decision-making to optimize traffic flow.

**Inferential Controls**: Edge devices are increasingly equipped with sophisticated inferential capabilities, allowing them to interpret environmental contexts accurately. These devices can communicate with broader infrastructural systems while making intelligent, contextually-aware decisions. In smart transportation scenarios, this translates to providing drivers with highly intelligent navigation instructions by integrating data from GPS and multiple camera inputs.

**Real-Time Data Analysis**: By enabling localized data analysis at the point of generation, edge computing significantly reduces information latency. This approach offers multiple advantages:

* Faster generation of actionable insights
* Reduced network bandwidth consumption
* Decreased operational costs
* Immediate decision-making capabilities

Across industries such as manufacturing, healthcare, telecommunications, and finance, edge computing facilitates more efficient and responsive IoT implementations.

**Enhanced Data Security**: By localizing data collection and analysis, edge computing inherently improves data security. Reduced extensive routing minimizes potential vulnerability points, making it easier to identify and mitigate suspicious activities before they escalate into significant security breaches. In fact, since edge computing provides computational resources, data are generated and processed within nodes in the edge network. This allows to remove private information before sending data to the cloud, avoiding potential privacy issues.

## Key requirements for successful deployment

As discussed by Hassan et al. [1], successful deployment of edge computing in IoT environment should meet specific requirements to achieve the different features provided by edge computing paradigm. Since several of these requirements are conflicting, application designers must find a good balance among all of them. These key requirements are:

**Latency Minimization**: Edge computing addresses the delay of traditional cloud models by processing data closer to its source, enabling real-time responsiveness. This is critical for applications like healthcare, autonomous vehicles, and industrial automation, where even minimal delays can have severe consequences.

**Reliability**: Reliable edge systems ensure consistent performance with minimal downtime, even in diverse and challenging conditions. This includes maintaining computational integrity and service quality while integrating with varied IoT ecosystems.

**Mobility Support**: As IoT devices increasingly operate on the move, edge computing must provide seamless connectivity and uninterrupted service. Robust mobility support ensures smooth handoffs, session continuity, and reliable performance in dynamic network environments.

**Real-time interactions**: Many IoT applications require immediate data processing and response. Edge computing supports real-time interactions essential for systems like collision avoidance in autonomous vehicles and industrial monitoring.

**Security**: Distributed edge systems introduce multiple vulnerability points. Comprehensive security measures, including data protection and continuous threat monitoring, are vital to safeguard against attacks while leveraging localized data processing.

**Interoperability**: Edge computing must integrate seamlessly across diverse hardware, protocols, and applications. Standardized interfaces and adaptive frameworks enable effective communication and collaboration in complex IoT ecosystems.

## Edge computing paradigms

Edge computing represents a decentralized computing platform that brings cloud computing capabilities closer to IoT devices at the network edge. Unlike traditional cloud computing, which relies on centralized remote servers, following Hassan et al. [1], edge computing offers several paradigmatic approaches:

* **Fog Computing**: Utilizes local network devices like routers or switches within a limited geographic region to provide computational services, emerging as a premier technology following IoT success.
* **Cloudlet Computing**: Involves performing delay-sensitive and computation-intensive tasks on local area network servers, reducing processing time for IoT devices.
* **Mobile Edge Computing (MEC)**: Brings cloud computing capabilities to the edge of cellular networks, with computational and storage services provided at base stations.
* **Mobile Ad Hoc Cloud (MAC)**: Leverages shared resources of available mobile devices in local proximity to process computation-intensive tasks.
* **Hybrid Computing**: Combines cloud and edge computing infrastructures to overcome cloud latency issues while maintaining access to large computing resources.

## Data types

Edge computing addresses various data types categorized primarily by their delay sensitivity:

* **Hard Real-Time Data**: Requires absolute zero tolerance for delay. These are critical data streams where even minimal latency can compromise system performance or safety.
* **Soft Real-Time Data**: Allows for bounded delays within specific time constraints. These data types can tolerate some processing time without significant performance degradation.
* **Non-Real-Time Data**: Represents delay-tolerant applications where immediate processing is not crucial, offering more flexibility in data handling and analysis.

The primary motivation for edge computing is its ability to handle these diverse data types more effectively than traditional cloud computing, especially for delay-sensitive applications.

## Applications

According to what discussed by Hassan et al. [1], edge computing has emerged as a transformative technology across multiple domains:

* Smart Homes: Enables automated monitoring and metering of utilities like water, electricity, and gas, allowing real-time data analytics at the network edge.
* Healthcare: Supports remote health monitoring, enables immediate reactions to health-related data, and resolves latency issues inherent in cloud-based medical applications.
* Video Surveillance: Facilitates efficient storage, management, and processing of video content from multiple cameras and sensors, often collaborating with cloud computing.
* Smart Grid: Supports real-time energy management by sensing consumption and distribution patterns, with edge computing providing agility and load distribution.
* Smart Cities: Assists in various urban management tasks such as street lighting control, environmental monitoring, emergency route exploration, and automated garden watering.
* Smart Logistics: Automates transaction flows between manufacturers and consumers, enhancing cost and time efficiency.
* Environment Monitoring: Enables comprehensive tracking of critical environmental parameters like gas concentration, water levels, soil humidity, and land position changes.

These applications demonstrate edge computing's potential to transform IoT by bringing computation closer to data sources, reducing latency, and enabling more responsive and intelligent systems.

# Use case: mobile game

Premsankar et al. [2] carried out an experimental evaluation of mobile gaming using a prototype edge computing platform. This experiment witnesses the power of edge computing deployment to achieve low-latency performances required by resource-demanding mobile games.

## Testbed setup

The researchers used the open-source GamingAnywhere cloud gaming platform to conduct their experiments. They focused on measuring the response delay, which is defined as the time elapsed between an action performed by the user and the occurrence of the corresponding outcome at the client device. The response delay includes three main components:

1. **Processing Delay (PD)**: The time taken by the server to process the user input and render the corresponding frame.
2. Testbed setup used for the network edge scenario.
3. **Playout Delay (OD)**: The time taken by the client to decode and display the frame on its own screen.
4. **Network Delay (ND)**: The round trip time (RTT) between the client and the server.

The experiments were carried out using a Google Nexus 5 mobile phone as the client device and a workstation with a 4-core Intel Xeon E3-1230 CPU, 16 GB of RAM, and two NVIDIA Quadro 2000 GPUs as the gaming server. In the experiments the authors considered two access technologies, i.e., Wi-Fi and LTE. The game of interest was Neverball, a game that is representative of a larger class of applications that rely on rendering complex 3-D environments, including virtual and augmented reality, and that need so a fast response to provide a smooth user interaction.

The researchers considered three different server deployment scenarios: 1) a local deployment at the network edge, 2) a special-purpose cloud computing infrastructure, and 3) a commercial public cloud provider.

## Experimental results

The researchers first studied the impact of server deployment on the network delay (ND). They found that the edge network scenario, with the server co-located with the LTE base station, achieved an ND of less than 20 ms, which is considered the state of the art of currently available wireless communication technologies. In contrast, the public cloud scenarios incurred significantly higher delays, at least twice as much as the edge network scenario.

Next, the researchers examined the overhead of different virtualization technologies (bare metal, container, and virtual machine) and how the screen resolution affected the response delay. They found that the performance of containers was almost the same as the bare metal configuration, while hypervisor-based virtualization incurred about a 30% higher processing delay (PD).

Finally, the researchers evaluated whether the use of more powerful computing resources in the cloud could compensate for the higher network delay. They found that the additional computational resources offered by the cloud were not effective for the full HD resolution, as most of the PD was due to the encoding of the source video content, rather than rendering.

## Implications and discussion

The researchers concluded that hosting computing resources very close to the end-users, at the access network edge, is the only viable option to achieve a satisfactory quality of experience for mobile gaming applications. While a response delay below 150 ms is generally considered acceptable for interactive applications, fast-paced interactions cannot tolerate delays beyond 70 ms. The edge network configuration allowed them to play the game at an HD resolution with processing times below 70 ms, which was not possible with the cloud-based deployments.

The researchers also discussed the implications of their findings for future IoT applications, highlighting the need for advanced virtualization, network function virtualization (NFV), and software-defined networking (SDN) techniques to handle the complexity of resource allocation and optimization in edge computing environments.

# Use case: industrial manufacturing

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

## Authors and Affiliations

**The template is designed for, but not limited to, six authors.** A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

### For papers with more than six authors: Add author names horizontally, moving to a third row if needed for more than 8 authors.

### For papers with less than six authors: To change the default, adjust the template as follows.

#### Selection: Highlight all author and affiliation lines.

#### Change number of columns: Select the Columns icon from the MS Word Standard toolbar and then select the correct number of columns from the selection palette.

#### Deletion: Delete the author and affiliation lines for the extra authors.

## Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is “Heading 5”. Use “figure caption” for your Figure captions, and “table head” for your table title. Run-in heads, such as “Abstract”, will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named “Heading 1”, “Heading 2”, “Heading 3”, and “Heading 4” are prescribed.

# Mobile edge computing to enhance low-latency IoT applications

## Figures and Tables

#### Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. 1”, even at the beginning of a sentence.

1. Table Type Styles

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

1. Sample of a Table footnote. (*Table footnote*)
2. Example of a figure caption. (*figure caption*)

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

##### Acknowledgment *(Heading 5)*

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

##### References

1. N. Hassan, S. Gillani, E. Ahmed, I. Yaqoob and M. Imran, "The Role of Edge Computing in Internet of Things," in IEEE Communications Magazine, vol. 56, no. 11, pp. 110-115, November 2018, doi: 10.1109/MCOM.2018.1700906.
2. G. Premsankar, M. Di Francesco and T. Taleb, "Edge Computing for the Internet of Things: A Case Study," in IEEE Internet of Things Journal, vol. 5, no. 2, pp. 1275-1284, April 2018, doi: 10.1109/JIOT.2018.2805263.
3. B. Chen, J. Wan, A. Celesti, D. Li, H. Abbas and Q. Zhang, "Edge Computing in IoT-Based Manufacturing," in IEEE Communications Magazine, vol. 56, no. 9, pp. 103-109, Sept. 2018, doi: 10.1109/MCOM.2018.1701231.
4. K. Zhang, S. Leng, Y. He, S. Maharjan and Y. Zhang, "Mobile Edge Computing and Networking for Green and Low-Latency Internet of Things," in IEEE Communications Magazine, vol. 56, no. 5, pp. 39-45, May 2018, doi: 10.1109/MCOM.2018.1700882.