Cluster Computing (2023) 26:2657–2680
https://doi.org/10.1007/s10586-021-03279-3

Improving latency in Internet-of-Things and cloud computing for real-time data transmission: a systematic literature review (SLR)

Saurabh Shukla[1] · Mohd. Fadzil Hassan[2] · Duc Chung Tran[3] · Rehan Akbar[4] · Irving Vitra Paputungan[5] · Muhammad Khalid Khan[6]

Received: 2 May 2020 / Revised: 17 March 2021 / Accepted: 29 March 2021 / Published online: 16 April 2021

Abstract

To store, analyse and process the large volume of data generated by IoT traditional cloud computing, is used everywhere. However, the traditional cloud data centres have their limitations to handle high latency issues in time-critical applications of IoT and cloud. Their applications are computer gaming, e-healthcare, telemedicine and robot surgery. The high latency in IoTs and cloud includes high computational, communication latency (service) and network latencies. The vital requirement of IoT is to have minimum network, service and computation latencies for real-time applications. Network latency causes a delay in transmitting a message or communication from one location to another. Services that require data in real-time are almost impossible to access the data via the cloud. Traditional cloud computing approaches are unable to fulfil the quality-of-service (QoS) requirements in IoT devices. Researches related to latency reduction techniques are still in infancy. Some new approaches to minimize the latency for transmitting time-sensitive data in real-time are discussed in this paper for cloud and IoT devices. This research will help the researchers and industries to identify the techniques and technologies to minimize the latencies in IoT and cloud. The paper also discusses the research trends and the technical differences between the various technologies and techniques. With the increasing interest in the literature on latency minimization and its requirements for time-sensitive applications; it is important to systematically review and synthesize the approaches, tools, challenges and techniques to minimize latencies in IoT and cloud. This paper aims at systematically reviewing the state of the art of latency minimization to classify approaches, and techniques. The paper uses a PRISMA technique for a systematic review. The paper further identifies challenges and gaps in this regard for future research. We have identified 23 approaches and 32 technologies associated with latencies in the cloud and IoT. A total of 112 papers on latency reduction have been examined under this study. The existing research gaps and works for latency reduction in IoTs are discussed in detail. There are several challenges and gaps, which requires future research work for improving the latency minimization techniques and technologies. Finally, we present some open issues which will determine the future research direction.

PRISMA: Preferred Reporting Items for Systematic reviews and Meta-Analyses

Keywords Internet-of-Things · Fog computing · Cloud computing · Latency · Service latency · Communication latency · Network latency · Queuing delay · Transmission delay · Computation latency

Abbreviations

IoT	Internet-of-Things
FIS	Fuzzy inference system
FC	Fog computing
MDP	Markov decision process
RL	Reinforcement learning
NN	Neural network
IDC	International Data Corporation
NFV	Network function virtualization
CDC	Cloud data centers
ICSN	Information-centric social networks
ICN	Information-centric network
VNF-RM	Virtual network function real-time migration
CDN	Content delivery network
VMM	Virtual machine migration
ECG	Electrocardiogram
SFC	Software function chaining
SPSRP	Service popularity-based smart resources partitioning
GAP	Generalized assignment problem
VNF	Virtual network function
EEG	Electroencephalogram

Extended author information available on the last page of the article

NP	Nondeterministic polynomial time
QoS	Quality of service
IRC	Information Resource Center
MS	Milli seconds
KB	Kilobytes
MB	Megabytes
KJ	Kilojoules
F-RAN	Fog-radio access networks
VM	Virtual machines
CBR	Constant bit rate
VBR	Variable bit rate
FCSS	Fog computing security service
CORD	Central Office Re-architected as a Datacenter
LR	Literature review
TCP	Transmission control protocol
EMG	Electromyography
KBPS	KiloBytes per second
RAM	Random access memory
WBAN	Wireless body area network
SDN	Software-defined network
PoP	Post office protocol
SFC	Software function chaining
WAN	Wide area network
LAN	Local area network
F2C	Fog-to-cloud
LOCPART	Latency optimized cache partitioning for cloud datacenters
IP	Internet Protocol
DCQCN	Datacentre quantized congestion notification
RDMA	Remote Direct Memory Access Technology
FDM	Frequency division multiplexing
MEC	Mobile edge computing
IRS	Intelligently reflecting surfaces
DDDPG	Double-duelling-deterministic policy gradient
DDQS	Double deep Q-learning scheduling
PFC	Priority-based flow control

1 Introduction

By 2020, IoTs in billions generating 507.5 zettabytes of data [1, 2]. Computing and transmission of this much data will certainly maximize the reply/service time of clouds. In recent years, the growth of IoT has caused major concern in services that require data in real-time. In the event of a huge transmission of data, the probability of error is directly proportional to it. Packet loss and transmission latency are directly linked to the data volume transmission from IoT to cloud servers. This can lead to poor QoS for end-users [3]. End-users suffered from unbearable high transmission latency and poor services due to the large volume of data transmitted from IoTs and this has posed a heavy burden on the cloud [4]. High service latency leads to the problem of synchronization between a request made by the client and the response given by the server; this generates very little service latency. Network congestion between IoTs and the cloud also lead to high latency. High service latency is the major cause of delay in most real-time services. The greater the distance, the greater number of gateway nodes are required for data packets forwarding to the end-point [5].

Furthermore, the process is delayed if the packet does not travel immediately through the router to reach the end-point, the packet may have to travel through many routers [6, 7]. Packets in routers normally suffered a delay of a few milliseconds i.e. per packet round trip time. These router gateway nodes add latency due to the underlying data communication delay [8]. Delays in answering the end-users queries also depend on the efficiency of the network and the quality of the routing equipment. There would also be some delays in the duplication of data in various cloud data centres located in different trans-continental regions [5]. These delays in response to high network latency is a major concern for IoT -cloud system, where the network state is constantly changing with time. Software applications that need to operate on trans-continental scales, or high-performance software systems need to operate on very fast response time, thus the network latency consumes a major part of the intended response time [5]. Uncertainty to latencies response is due to network unpredictable condition. When the data is transmitted from IoT to cloud. The data again goes for a new life cycle on the cloud servers. The energy of the sensor gets reduced by the large amount of data stored on them [9]. Secure storage is the requirement of sensor nodes. Therefore, there is scope for working in this domain. The study aimed at filling that gap by conducting a literature review of the approaches, tools, challenges, techniques and technologies. This study provides an in-depth understanding of the latency minimization techniques and technologies in IoT and cloud. The goal of this paper is to give an inside observation of the latency issue in IoTs and cloud with comparison of the other known latency reduction techniques. The aim is to motivate future works to develop a latency aware model for IoT, cloud and end-users. The contributions of this study work can be summarized as follows:

(A) A classification of the reported techniques and technologies.
(B) Present the research trends in latency reduction techniques and technologies by investigating the

number of published researched works and search occurrences in google scholar.

(C) Review of several latency reduction techniques and technologies.

(D) Identify latency reduction techniques research gaps in IoT, cloud and fog computing.

(E) Address the limitations of current research works and some open issues on latency requirement for time-sensitive applications. From this survey, the researchers and industrialists will be able to gain an insight view of IoTs and cloud latency requirement for time-sensitive applications with a better understanding of latency reduction techniques and technologies.

It should be mentioned here that several works have been done to improve latency, response time and storage in IoT and cloud, whereby the most relevant published works are included in this paper (Tables 1, 2).

2 SEARCH

Popular literature resources are utilized for this chapter. The search data for this chapter is based upon the following digital libraries:

1. ACM Digital Library (http://dl.acm.org)
2. IEEE Xplore Digital Library (http://ieeexplore.ieee.org)
3. Science Direct (http://www.sciencedirect.com)
4. Springer Link (http://link.springer.com)
5. Wiley Online Library (http://onlinelibrary.wiley.com)
6. Web of Science (https://apps.webofknowledge.com)
7. Scopus (https://www.elsevier.com/solutions/scopus)

We have utilized several keywords to perform the search process.

The search process is carried out with two kinds of operators i.e. AND and OR. The result collected from AND operator is not enough, so the OR operator is also used.

Title -ABS-Key: ((“Cloud computing” AND “Latency”, “Internet-of-Things” AND “Fog computing”, “Healthcare Internet-of-Things” AND “Fog computing”, “Communication latency” AND “Service latency”, “Computation latency AND “Network latency”, Communication latency” AND “Service latency” AND “Computation latency AND “Network latency”, “Response time” AND “Fog”, “Fog nodes” AND “Cloud”)).

Title -ABS-Key: ((“Cloud computing” OR “Latency”, “Internet-of-Things” OR “Fog computing” OR “Communication latency”, “Service latency” OR “Computation latency, “Network latency” OR “Response time” OR “Fog”, “Fog nodes” OR “Cloud”)).

See Fig. 1 Phases of the search process.

See Fig. 2 shows the SLR overview.

See Fig. 3 for the types of papers included specifically for latency minimization.

3 Current latency minimization technologies and techniques for real-time data transmission

There are a total of 32 technologies and techniques discussed in this section for latency minimization. These technologies are used in cloud computing and IoT devices. The selections of these techniques are based on the coverage/performance metrics and approach used by them for latency minimization. Covering features or performance metrics for comparing various latency reduction techniques are:

1. Latency.
2. Execution time
3. Response time.
4. Throughput.
5. Network Traffic.

3.1 Discussion

These 32 technologies and techniques have different approaches to reduce latency in cloud and IoT devices to transfer the data in real-time. There is a total of 23 approaches used by these techniques. These approaches are of types unary, binary and hybrid. The hybrid approach

Table 1 Inclusion and exclusion criteria

Inclusion Criteria (I1, I2, I3)
I1: Paper that explicitly discusses latency in cloud and IoT
I2: Paper that focuses on how FC is advantageous for latency minimization
I3: Papers that are focused on presenting challenges associated with IoTs and cloud
Exclusion criteria (E1, E2)
E1: Papers that do not have latency in cloud and IoT as a primary study
E2: Papers that are not concerned with FC as a primary source of study
E3: Papers that are not concerned with the challenges related to IoT and cloud

Table 2 Comparison of this SLR with existing review articles

Study	Focus	#Included papers	Search date
T. Bai et al.	IoT and Mobile edge computing	46	2020
J. Jiang et al.	IoT	90	2020
R. O. Aburukba et al.	IoT	48	2020
Z. Chang et al.	IoT	27	2020
E. Navarro et al.	IoT	201	2020
M. M. Martín-Lopo et al.	IoT	110	2020
A. Pliatsios et al.	IoT	62	2020
P. Bellavista et al.	IoT and fog computing	133	2019
R. K. Naha et al.	Fog computing	142	2018
M. Mukherjee et al.	Fog and cloud computing	190	2018
L. Bittencourt et al.	IoT, fog and cloud computing	171	2018
A. Brogi et al.	IoT and fog computing	25	2017
C. Mouradian et al.	Fog and cloud computing	169	2017
O. Osanaiye et al.	Cloud and fog computing	122	2017
S. B. Baker et al.	IoT	95	2017

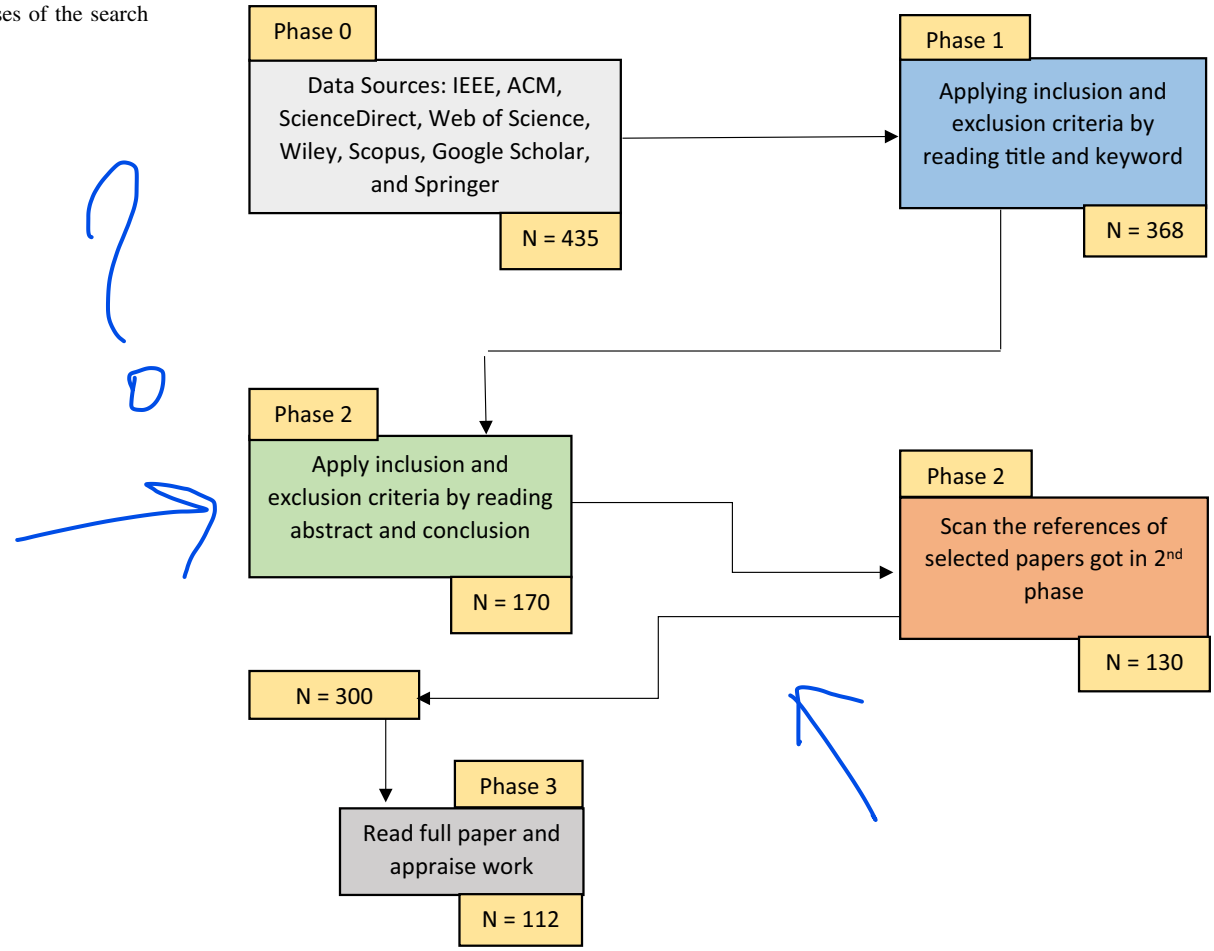


Fig. 1 Phases of the search process

uses multi techniques to reduce latency for the transmission of data in real-time. Whereas, binary type approach uses at most two technique and unary approach use only one technique for latency reduction in IoT devices and clouds.

Tables 3, 4, 5 and 6 in this chapter discuss the different technologies and techniques along with the approach used and the description of their work. There are time-sensitive applications and emergency responses that require low

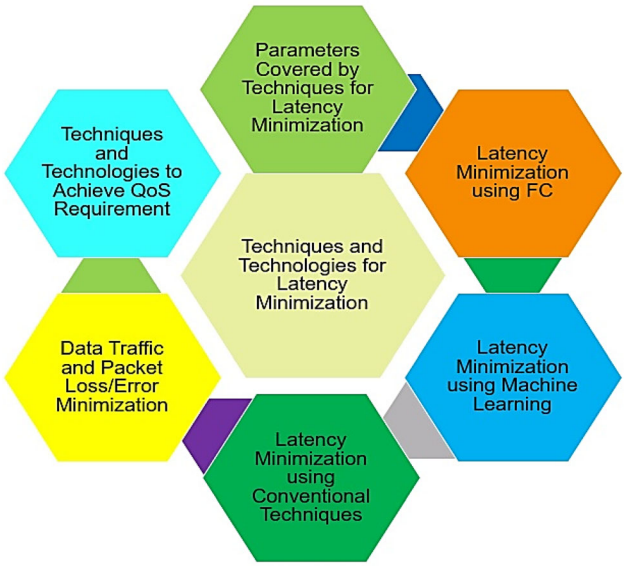


Fig. 2 SLR overview

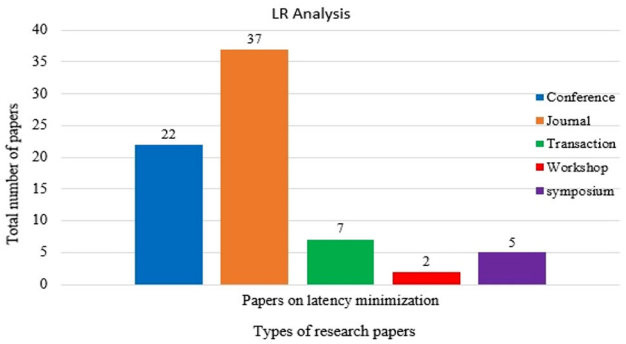


Fig. 3 SLR analysis for latency inclusion articles

latency, and delay caused by transferring data to the cloud and then back to the application can seriously impact their performances. Transmission and analysis of a large amount of data will increase the response time of cloud computing. Here FC technology covers all the parameters required for latency minimization and transfers the data in real-time.

See Table 7 shows the research questions and motivations related to this study.

4 Existing research work related to research questions

This section discusses the various existing work focusing on IoTs, cloud, and FC concerning high latency, high network usage, and high bandwidth consumption. This section presents the in-depth analysis and the comparison of existing works.

FC = FOG COMPUTING

RQ1 How to control or minimize the high latency between IoTs and cloud?

Fog nodes can transfer the data to other fog nodes using data packet allocation through machine learning; the data can directly be sent to the end-users; this method of data packet allocation will minimize the high latency. FC is smart enough to decide how to send when to send and what to send in a real-time environment [19, 41]. Analyzing the most time-sensitive data at the network edge, close to where it is generated instead of sending vast amounts of IoT data to the cloud.

(1) Acting on IoT data in milliseconds, based on policy.
(2) Sending selected data to the cloud for historical analysis and longer-term storage.

In January 2014, Cisco launched its vision of FC, designed with the idea of bringing cloud computing facilities and network side capabilities.

4.1 Latency minimization using FC

This section discusses the various existing techniques and algorithms for latency minimization in IoT and cloud using FC.

1. *iFogStor* In [15], the authors discussed the issue of latency for IoTs. A technique called iFogStor was proposed using FC. iFogStor used the heuristic approach. Its performance was much better as the latency was reduced by more than 86% as compared to the cloud and 60% when compared to fog computing solutions. Future work requires an architecture to efficiently handling time-sensitive healthcare IoT applications.
2. *FC-Cloud fusion* In [42], the authors discussed various aspects and research challenges associated with the fusion of fog and cloud for latency minimization, resource allocation, optimization, energy consumption, and RAM usage in IoTs. They expressed the problem of an increase in the number of heterogeneous devices and applications.
3. *F-RAN* In [29], the authors proposed an F-RAN model called Fog-Radio Access Network to minimize latency in IoTs and wireless devices. They proposed an algorithm for latency driven applications to provide services to multiple end-users with heterogeneous resource allocation. The algorithm helps in sharing the resource among several users.
4. *Live VM migration* In [43], the authors reviewed a research work from cloud computing to FC and introduced a conceptual live VM Migration framework to minimize service latency in IoT-fog-cloud infrastructure. A conceptual live approach for VM

Table 3 Technologies and techniques for latency minimization

Technologies and techniques (T_i (i = 1, 2…. n)
T_1. Reinforcement learning code offloading mechanism [10]
T_2. Hermes [11]
T_3. Hipster [12]
T_4. Content Delivery Network (CDN) [13]
T_5. Cloudlet [8, 14]
T_6. iFogStor [15]
T_7. Multipath TCP (MPTCP) [16]
T_8. Femto cloud [17]
T_9. Home cloud [18]
T_{10}. SDN [18, 19]
T_{11}. CORD [6]
T_{12}. NEBULA [6]
T_{13}. Computation Offloading [20, 21]
T_{14}. VMM [22]
T_{15}. Data Virtualization [23, 50]
T_{16}. CDC [23]
T_{17}. FC [24]
T_{18}. Binary ATM [25]
T_{19}. Locpart (Latency optimized cache partitioning for cloud data centres [26]
T_{20}.VNF Real-time Migration (VNF-RM) [27]
T_{21}: FogPlan [28]
T_{22}: FogTorch [4]
T_{23}. F-RAN [29]
T_{24}. Healthfog [30]
T_{25}. FogBus [31]
T_{26}. Markov Decision Process (MDP) [32]
T_{27}. Orthogonal FDM [33]
T_{28}. NB-IoT [34]
T_{29}. Intelligently Reflecting Surfaces (IRS) [35]
T_{30}. Mobile Edge Computing (MEC) [35]
T_{31}. Remote Direct Memory Access technology (RDMA) [36]
T_{32}. Non-orthogonal multiple access [37]

migration is proposed to minimize the downtime and migration time to guarantee and avail the resource, services to end-users. The research work lacks the real-world implementation and deployment of the framework.

5. *Many-to-one matching* In [14], the authors formulated the problem of optimization to select cloudlets and minimize latency in IoTs and fog networks by controlling and monitoring the workload. A rankings system was introduced for many-to-one matching IoT nodes with cloudlets. An algorithm was also proposed to solve the matching game between IoT and cloudlet.
6. *FCSS* In [44], the authors highlighted the issues and requirements for social networks linked to the information centre. Their requirement consists of ultra-low latency. An FC-based security service issued for this purpose. Here, the large data transmission at the edge of the networks leads to the problem of high data traffic and network congestion which generates high network latency.
7. *Cost-effective schema* In [45], the authors discussed the issues related to hardware and, software failure, high latency with limitations of resources in different neighbouring fog nodes. The authors further proposed a deployment schema for the services between IoT, fog and cloud. Their discussed work is thought of as the most cost-effective and scalable. But no work was done to minimize the high latency for SFC over fog nodes and cloud servers despite the authors did mention that high latency is a critical issue for IoTs deployment.
8. *SPSRP* In [46], the authors discussed the problem of resource utilization at the individual fog nodes.

Table 4 Technologies and techniques (T_i (i = 1, 2…n) to minimize the delay in IoT devices and cloud for transmission of data in real-time

T_i	Latency	Execution time	Response time	Throughput	Network traffic
T_1	✔	✔	✔		✔
T_2	✔	✔	✔	✔	
T_3	✔		✔		✔
T_4	✔			✔	
T_5	✔		✔	✔	✔
T_6	✔	✔	✔	✔	✔
T_7	✔		✔		✔
T_8	✔				✔
T_9	✔	✔			✔
T_{10}	✔		✔	✔	✔
T_{11}	✔			✔	
T_{12}	✔		✔		✔
T_{13}	✔		✔		✔
T_{14}	✔			✔	✔
T_{15}	✔	✔			
T_{16}	✔			✔	✔
T_{17}	✔	✔	✔	✔	✔
T_{18}	✔	✔			✔
T_{19}	✔		✔		
T_{20}	✔			✔	✔
T_{21}	✔		✔	✔	✔
T_{22}	✔	✔	✔	✔	✔
T_{23}	✔	✔	✔		✔
T_{24}	✔	✔			✔
T_{25}	✔	✔		✔	✔
T_{26}	✔		✔		✔
T_{27}	✔	✔		✔	
T_{28}	✔	✔			✔
T_{29}	✔		✔	✔	✔
T_{30}	✔	✔			✔
T_{31}	✔			✔	✔
T_{32}	✔	✔		✔	

The covered parameters or performance metrics are listed in the below table

There was an issue related to coordination between computational efficiency of different fog nodes. This leads to heavy traffic and network congestion. The authors a resource partitioning method for IoTs and FC. Their proposed research work reduces the response time in IoTs However, there is an issue of high network latency and high computational latency in IoTs.

9. *Cloud-fog based service* In [47], a cloud-fog based service was proposed to minimize latency; data transmission and traffic generated due to heterogenous applications. Cloud unable to meet the QoS requirement. There was a lack of work done in minimizing high computational latency for healthcare applications. However, there was a lack of work done in minimizing high computational latency for healthcare applications.
10. *SDN* In [48], the authors proposed an Software Defined Network (SDN) for IIoTs. A system for task allocation. Using the task priority, real-time performance is achieved using FC. Their proposed method was able to minimize latency in IoTs. However, the research work lacks providing computing services for mobile devices.
11. *Load-balancing* In [49], the authors proposed a load-balancing algorithm for IoT-fog-cloud system. The algorithm works for multiple gateways. A queueing model was employed to measure the latency in data transmission from IoT-fog network. Furthermore,

Table 5 Approach and its type used by different techniques

Approach (A_i) (i = 1, 2…n)	Approach Type
A_1. RL [T_1] [T_3] [T_5] [T_{15}]	Hybrid
A_2. Distributed Network [T_1] [T_4] [T_6] [T_8] [T_{11}] [T_{13}] [T_{14}] [T_{15}] [T_{19}] [T_{21}] [T_{23}]	Hybrid
A_3. Multi-Agent System [T_5] [T_{19}]	Binary
A_4. VMM [T_3] [T_7] [T_{20}] [T_2]	Hybrid
A_5. Parallel Processing [T_4] [T_{19}]	Binary
A_6. Heuristic Learning [T_3] [T_5] [T_6] [T_8] [T_{10}] [T_{21}] [T_{22}] [T_{23}]	Hybrid
A_7. Batch Processing [T_5] [T_{18}] [T_2]	Hybrid
A_8. Intelligent Caching [T_6]	Binary
A_9. SDN [T_{11}] [T_{13}]	Binary
A_{10}.Virtualization [T_{11}] [T_{13}] [T_{14}] [T_{19}] [T_{23}] [T_{24}] [T_{25}]	Hybrid
A_{11}. Multitenancy [T_{12}]	Unary
A_{12}. NFV [T_{13}]	Unary
A_{13}. Random Forest [T_{15}]	Unary
A_{14}. Block Migration [T_{16}] [T_9] [T_2]	Hybrid
A_{15}. Data Integration [T_{20}]	Unary
A_{16}.Greedy algorithm [T_{21}] [T_{22}]	Binary
A_{17}. FIS [T_{19}]	Unary
A_{18}. NN [T_{25}]	Unary
A_{19}. Lightweight pre-scheduling algorithm [T_{17}]	Unary
A_{20}. Double-Duelling-Deterministic Policy Gradient (DDDPG) algorithm [T_1] [T_{17}]	Binary
A_{21}. Double deep Q-learning scheduling algorithm (DDQS) [T_{17}]	Unary
A_{22}. Approximation algorithm [T_{30}]	Unary
A_{23}. Job-scheduling algorithm [T_{15}] [T_{16}] [T_{25}] [T_{26}]	Hybrid

Hybrid: More than two techniques

Binary: Two techniques

Unary: Single technique

multiple gateways are employed to increase the performance of the network.

12. *Fog infrastructure* In [50], the authors address the design and dimensions of a fog infrastructure using an interlinear programming approach to minimize the high latency and data traffic for time-sensitive applications in IoT networks. They also proposed a column generation model for QoS requirement in IoT.
13. *Lightweight pre-scheduling algorithm* In [51], the authors proposed a novel IoT application placement in fog environment to minimize the high latency, execution time, and energy consumption. Furthermore, they proposed a weighted cost model with multiple IoT devices. Their proposed technique was based on the memetic algorithm which helps in making decisions for batch applications. They also prospered a lightweight pre-scheduling algorithm considering the heterogeneous IoT applications to maximize the concurrent execution for the number of parallel tasks.
14. *Joint computation offloading* In [52], the authors proposed a dynamic optimization scheme to minimize service latency and energy consumption using FC in IoT networks. They proposed a scheme that performs dynamic resource allocation and computation offloading with multiple devices present at the edge. Furthermore, they have used a join computation offloading and radio resource allocation where the main problem was divided into several sub-problems.
15. *Fog-based architecture* In [53], the authors presented a fog-based architecture for healthcare IoT system. They highlighted the issue of high latency, end-to-end delay and high bandwidth consumption in IoT-cloud system. Fog nodes work at the edge of networks i.e. at the LAN. The fog servers are further partitioned into several virtual machines. Furthermore, they have used the Elliptive curve cryptography technique to authenticate the patients or end-users.
16. *FC-based architecture* In [54], the author proposed an FC-based approach for minimizing computational latency in mobile edge devices. They have proposed an FC-based architecture for resource allocation and computational offloading. Furthermore, they have

Distrib
uted

Table 6 List of technologies and techniques along with their description

Technologies and Techniques (T_i) (i = 1,2,3…. n)	Description (D_i) (i = 1,2,3…n)
T_1. RL code offloading mechanism	D_1. Mobile services in real-time
T_2. Hermes	D_2. Novel Fully Polynomial Time-Approximation Scheme (FPTAS)
T_3. Hipster	D_3. Workload Distribution
T_4. CDN	D_4. Transfer the static data in real-time mode
T_5. Cloudlet	D_5. Resource–intensive and interactive mobile applications
T_6. iFogStor	D_6. Latency reduction by 86% and 60% as compared to cloud and fog
T_7. MPTCP	D_7. It offers improvements in latency as compared with TCP
T_8. Femto Cloud	D_8. Enables multiple mobile devices to be configured into a coordinated cloud computing service
T_9. Home cloud	D9. A kind of personal cloud that reduces data transmission risk
T_{10}. SDN	D_{10}. Identification of low-latency packets
T_{11}.CORD	D_{11}. It provides mobility and supports location awareness
T_{12}.NEBULA	D_{12}.Supports local awareness
T_{13}.Computation Offloading	D_{13}. Deployment of multiple mobile agents was done to find the best suitable options to offload
T_{14}.VMM	D_{14}. Load balancing across multiple locations
T_{15}.Data Virtualization	D_{15}. Provides direct access to live operational data
T_{16}. CDC	D_{16}. Transfers and loads copies of the changes into the target data warehousing system
T_{17}. FC	D_{17}.Acts on IoT data in milliseconds (ms)
T_{18}.Binary ATM	D_{18}.Node processing delay is less in binary ATM when compare with conventional ATM
T_{19}.Locpart(Latency optimized cache partitioning for cloud)	D_{19}. Locpart guarantees QoS and improves global system performance
T_{20}.VNF Real-time Migration (VNF-RM)	D_{20}. Reducing the network latency by 70% to 90% after latency aware VNF migrations
T_{21}. FogPlan	D_{21}. Use a heuristic approach to minimize latency
T_{22}. Fog Torch	D_{22}. Use of Java Tool to meet the QoS requirements of IoTs
T_{23}. Fog-RAN	D_{23}. Use a heuristic algorithm to minimize latency
T_{24}. Healthfog	D_{24}. Used the FC approach with neural network algorithm to meet the QoS requirements of healthcare IoT
T_{25}. FogBus	D_{25}. A framework to integrate IoT-Fog (Edge)-Cloud
T_{26}. MDP	D_{26}. Used for the stochastic environment and decision making
T_{27}. Orthogonal FDM	D_{27}. Used a group-based service to achieve ultra-low latency
T_{28}. NB-IoT	D_{28}. Used in a dynamic system to achieve minimum latency
T_{29}. IRS	D_{29}. Used computation offloading mechanism in the MEC system
T_{30}. MEC	D_{30}. Used at the edge of networks for latency minimization
T_{31}. RDMA	D_{30}. Used direct memory access for the HPC system
T_{32}.Non-orthogonal multiple access	D_{30}. Used short packet communication to achieve low latency

discussed several issues in mobile edge devices such as packet loss, packet error, network failure, and energy consumption.

17. *FC-based IoT architecture* In [55], the authors conducted an in-depth analysis of FC and its role in IoT networks. Detailing how FC can be used to at the edge of networks. Next, they introduce an FC-based architecture for IoT. They have highlighted several issues in IoT such as high latency, energy consumption, computational latency, and network latency.

4.2 Latency minimization using machine learning

This section includes the existing machine learning techniques and algorithms to minimize the high latency between IoT and the cloud.

1. *Hipster* To fulfil the requirement for QoS, the authors in [12], introduced a solution called Hipster. It is a combination of RL and a heuristic approach. It assists in improving resource efficiency, and

Reinforcement Learning

Table 7 Research questions associated with background and related work

Research question	Motivation
RQ1. How to control or minimize the high latency between IoT and cloud?	In many time-critical applications of IoT, cloud-scale processing and storage are not required. Extreme time-bounded selection should be made on things that produce and act on the data [38]. Few IoTs have time-sensitive applications that require transferring of data in real-time [39]
RQ2. How to control or regulate the high data traffic rate and minimize the path loss and packet error/loss between IoT and cloud?	By 2020, there will be around 50 billion connected devices, which will generate 507.5 zettabytes of data by the end of the decade [40]. 30.7% of the IoT devices will be found in healthcare. By increasing transmission and determination of these high volumes of data, cloud computing relative reply time will rise. The path loss, packet loss, and transmission latency are proportional to the amount of data transmitted from IoT to cloud servers [38]
RQ3. How the network of IoT and cloud can meet the QoS requirements for time-sensitive applications?	In the case of large data transmission, the more data is transmitted over a network, the higher will be the probability of error occurs. This leads to poor QoS for end-users [9]. Healthcare IoT applications require the data in minimum time and with low latency. Traditional cloud servers are unable to fulfil the QoS requirements of healthcare IoT [7]

mapping between latency demanding tasks and batch workloads. Their proposed technique improves resource adequacy, throughput, and response time and it supports batch processing. Hipster deployed Hipster's Heuristic Policy to allow collaboration of latency demanding and batch workloads in common and mutual data centres. However, the authors did not discuss the problem of high communication latency between the IoT devices, end-users, and cloud servers.

2. *Hermes* In [11], the authors proposed a technique to reduce latency for mobile computing applications. The main function of the proposed technique includes the optimization of task assignment for devices that are deprived of resources. The proposed research work focused on the offloading of computational tasks. Using this machine learning technique, the authors proposed the formulation of an NP-hard problem approach to further decreased the latency. The authors did not discuss the problem of round-trip time delay which related to high communication latency between IoT devices, mobile users and cloud servers. However, the result showed that the authors managed to minimize high network and high computation latency with a greater number of percentages as compared to other existing works in this area.
3. *Basic block offloading* The authors in [10], used an RL technique to offload the blocks in the FC-mobile system. The technique was proposed for a distributed system. The proposed work minimizes the latency and computation time in a multi-agent environment.
4. *Fuzzy-based model* The authors in [56], introduced a model-based of Fuzzy inference system for VANETs. The proposed model was able to minimize the latency in VANETs. To make the decisions in real-time rules were made using fuzzy for the proposed model in VANET. The model was used to check and track the experience; further checks the accuracy of the events occurring in the system. The research work lacks real-world implementation to validate the proposed model.
5. *Hybrid bio-inspired algorithm* The authors in [57], developed and implemented a bio-inspired algorithm to reduce the high in IoT-FC-Cloud system. The algorithm is a hybrid of particle and cat swarm optimization. It manages the resources and schedules the task at the fog nodes. RL here can be used as a future work for resource allocation and task management in the IoT-FC system.
6. *Genetic algorithm* In [58], the authors used a genetic algorithm as a heuristic approach for scheduling the IoT service request. This helps in achieving minimum latency for IoT-fog-cloud hybrid network. They highlighted the issue of the NP-hard nature of scheduling problem in IoT network. Their proposed method was to minimize the queuing delay in the IoT-fog system. Furthermore, they have used integer programming to minimize the overall service request latency in IoT.
7. *Blockchain-based architecture* In [59], the authors proposed a blockchain-based intelligent IoT architecture. The proposed architecture was able to minimize the latency in IoT. Furthermore, their

GOOD.

Fluke

Best, but no blockchain

proposed work was able to provide secure data transmission between IoT and the cloud. The architecture was divided into two parts qualitative and quantitative analysis. It was an AI-driven IoT architecture which works both for security analysis and latency minimization in IoT networks.

8. *Double-Duelling-Deterministic Policy Gradient* In [60], the authors used computation offloading with deep reinforcement learning algorithm for QoE (Quality-of-Experience) to minimize high latency in IoT networks. A QoE model was proposed to minimize the service latency, transmission latency, energy consumption, and transmission computation. They proposed a Double-Duelling-Deterministic Policy Gradient (DDDPG) algorithm by improving the Deep Deterministic Policy Gradient (DDPG) algorithm for latency minimization at the edge of IoT networks. The algorithm performs better than the existing works.
9. *Heuristic approach* In [61], the authors used an exhaustive and heuristic approach to minimize the latency attacks on IoT. This further minimizes the delay in the transmission of data. Their proposed model was able to achieve maximum stability under latency attack by conducting parameter tuning.
10. *Fuzzy-based approach* In [62], the authors proposed a fuzzy-based approach with a mobile edge orchestrator to minimize latency in IoT by splitting the task and to avoid task allocation failures at the edge of the network. Their proposed work was an optimal task execution and task handling strategy for avoiding the high latency in data transmission. They further perform data offloading mechanism in their approach.
11. *Visible light communication architecture* In [32], the authors proposed a visible light communication architecture to fulfil the QoS requirement of IoT. They further used the Markov Decision Process (MDP) of the reinforcement learning algorithm to solve the issue of uplink and downlink energy resource management for latency minimization in IoT. The results generated from their prosed work outperforms the other existing algorithms working in this issue.
12. *Fog-based resource allocation* In [63], the author proposed a fog-based resource allocation technique using a machine learning algorithm for latency, energy, network usage minimization in IoT. A detailed in-depth analysis was conducted for the request of IoT applications such as low service latency, and low computational latency. They have used reinforcement learning algorithm to minimize the latency and meet the QoS requirement for IoT.
13. *Double Deep Q-learning algorithm* In [64], the authors proposed a double deep Q-learning scheduling algorithm (DDQS). The algorithm works on the task scheduling of IoT devices at the fog node. The proposed algorithm was able to minimize the service latency, cost, computational latency, energy consumption. Furthermore, it also handles the Point of Failure (PoF), and load balancing issue.

4.3 Latency minimization using conventional techniques

This section includes a detail description, discussion, and analysis of existing conventional techniques used in IoT and cloud for latency minimization.

1. *CORD* In [6], the authors discussed the concept of CORD for minimizing latency, virtualization at the edge, delivering end to end services with Software Defined Network (SDN), Network Function Virtualization (NFV). It integrates NFV, SDN and cloud. CORD uses the virtualization technique and works as a data centre. CORD provides mobility support for the IoT devices and provides SDN at the edge. They also discussed the concept of NEBULA for latency minimization and location awareness in IoT. It is used for distributed data-intensive applications such as MapReduce. Nebula works at the edge of networks. However, it is not suitable for a centralized computing environment.
2. *Femto cloud* In [17], the authors proposed the method of the Femto cloud. Their proposed method specifies a changing, self-configurable and multi-device mobile cloud from a group of mobile devices. This method enables many mobile devices to be arranged in integrated cloud computing servicing. The scheduler necessarily appoints duties using scheduling algorithm accessible tools to increase the available metrics while managing device churning.
3. *Home cloud* In [18], the authors used the technique of Home cloud for device automation, IoT application delivery services, automated orchestration and minimizing latency. Home cloud supports virtualization and SDN, multiple apps on the same infrastructure and support network function. It is a kind of personal cloud that reduces data transmission delay.
4. *IoT issues* In [65], the authors discussed the various issue which affects the latency in the IoT network such as large data transmission, high data traffic, network usage, bandwidth consumption and queueing delay. They further discussed the challenges and issues related to QoS requirement for IoT. Moreover,

they discussed the role of fog and cloud computing to achieve the minimum latency in IoT.

5. *Orthogonal FDM* In [33], the authors proposed a group-based service using orthogonal Frequency Division Multiplexing (FDM) to provide ultra-low latency for uplink IoT networks. Their proposed model works on different varied time frequencies and supports massive communication and connectivity with reliable ultra-low latency. The system was also useful for massive machine system.
6. *Approximation algorithm* In [66], the authors proposed an approximation algorithm to solve the issue of high latency in IoT and MEC using NP-hard. They further decompose the problem into three sub-problems. Their proposed algorithm was able to find the optimal solution. The results generated proofs of superiority with other existing work.
7. *Resource allocation* In [67], the authors proposed a resource allocation method for secure communication in IoT with ultra-low latency. Their proposed method was able to maximize throughput and minimize power consumption. The method was efficient to provide the optimal solution for the problem related to high latency in IoT.
8. *NB-IoT* In [34], the author presented an NB-IoT system to monitor electrical appliances in a smart grid network with the requirement of minimum latency for internet-connected devices. It was a dynamic system that can monitor maximum latency of 8 s for indoor and 2 s for outdoor. Their proposed system was an efficient system for smart grid and IoT network latency requirement.
9. *Resource allocation with NB-IoT* In [68], the authors perform the resource allocation for NB-IoT to analyze and measure the rate of change of latency. The proposed algorithm was able to allocate the resources at the uplink transmission and manage to further configure it. Furthermore, the performance is measured in terms of latency, rate and power. The proposed work was able to outperform the round-robin and brute-for approach when compared for latency minimization in IoT.
10. *Task allocation approach* In [69], the authors proposed a novel task allocation approach to minimize latency in the IoT network. They further form the groups of the nodes at the edge of the network to maintain the reliability of the network by efficient fault tolerance. The author was able to increase task reliability. Their proposed technique works in a decentralized manner. Finally, they were able to minimize the latency for task allocation at the edge.
11. *Intelligently Reflecting Surfaces (IRS)* In [35], the authors used computation offloading mechanism in Mobile Edge Computing (MEC) system for resource-intensive and time-sensitive applications. Furthermore, their proposed method was able to minimize latency in MEC using IRS.
12. *Datacentre Quantized Congestion Notification (DCQCN)* In [70], the authors proposed a predictive Priority-based Flow Control (PFC) using DCQCN for reducing tail latency, response time and energy consumption. They have used Remote Direct Memory Access technology (RDMA).
13. *Direct memory access* In [36], the authors proposed a high-performance communication system for a database system based on a direct memory access technique for the intra-data centre (RDMA) and intra-machine communication. RDMA and Ethernet were able to minimize latency and bottleneck in the network.

RQ2 How to control or regulate the high data traffic rate and minimize the path loss and packet error /loss?

The below discussion on the related work highlights the issue of high data traffic, path loss and packet error in IoTs and cloud. This section highlights the issue of high data traffic and packet error/loss in IoT and the cloud. A detailed discussion and analysis of existing research work are done to minimize data traffic and packet loss using FC and conventional techniques.

1. *Computation Offloading* In [21], the authors proposed a resource optimization and resource allocation framework. The resource allocation can be done by partitioning the data packets into small chunks. These chunks are further allocated to the other participating nodes. This process is called computation offloading here the task is distributed to other sub clouds called cloudlets by deciding on the size of the task and the number of tasks waiting in a queue. This process will help in load balancing among the nodes. This was an optimal offloading decision. The proposed method identifies the resource capabilities of nodes to minimize the overall data traffic rate. The user can decide to offload the data by its own rules and information.
2. *Cloudlet* Similarly, in [71] a cloudlet was proposed to minimize the high data traffic between edge device and cloud. The cloudlet consists of various virtual machines. The machines were designed to migrate from one cloudlet to another. The author claims that the application runs faster with this type of configuration. However, the author faces challenges due to the large response time of the application.

3. *FC framework* In [72], the authors proposed an FC framework for IoT and cloud. They also proposed architecture for minimizing latency, discussed several challenges and optimization. They further discussed resource allocation techniques to minimize the load burden and network traffic using FC. However, the research work lacks real-world implementation.
4. *Resource allocation* In [73], the authors proposed an algorithm to solve the problem of resource allocation in IoT-fog-cloud infrastructure. The algorithm works on a handover scheme for mobile IPV6 and uses scheduling policies which further reduces the network usage, data traffic, response time and latency in IoTs. But their research work lacks the work on minimizing the RAM consumption for process execution.
5. *IoT and cloud interconnection* In [74], the authors performed a detailed and comprehensive analysis of the relations between IoTs, edge computing and cloud computing using FC. They further discussed how these existing and emerging technologies can minimize the high data traffic and latency between IoT devices, end-users, and the cloud. But the reliability associated with emergency response application services is major concern which leads to long delays and data loss.
6. *Collaborative communication* In [75], the authors discussed the concept of minimizing latency and network traffic in the IoT-cloud environment using FC. They proposed an algorithm to have collaborative communication between fog nodes. This helps in minimizing end-to-end latency and data traffic in IoT-fog-cloud infrastructure. Their research work lacks real-world implementation to validate the proposed model.
7. *Computation offloading and FC architecture* In [20], the authors used the concept of computation offloading with a hybrid method to minimize energy consumption and network traffic in IoTs using FC. In [76], the authors discussed the FC architecture for latency and network traffic minimization in IoTs. Furthermore, they presented a taxonomy for FC for IoT.
8. *IoT framework* In [39], the authors introduced the framework for IoT-fog-cloud. They also proposed an analytical model to reduce the service delay in IoTs. The authors further discussed the FC approach to implements the idea of extending the cloud services at the edge of networks to improve system performance and resource efficiency. The proposed framework reduces network bandwidth consumption and data traffic in the network.
9. *Survey and analysis on FC* In [40], the authors presented a detailed survey on FC to minimize the network traffic and latency in IoTs. The authors also discussed the network application, research challenges and fundamentals of FC in connection with IoT and the cloud. Moreover, FC has the potential to increase the performance and efficiency of IoT devices. Similarly, the authors in [41] done a detailed analysis of research challenges associated with FC. The survey analysis consists of FC architecture for latency minimization and an algorithm for network traffic control. However, the research work lacks real-world implementation.
10. *Virtual fog framework* The authors in [77] proposed a virtual fog framework. A network function virtualization (NFV) was proposed to describe the virtual fog in the IoT system. This virtual technique in FC was able to minimize the delay, data traffic and jitter. However, resource utilization is a major problem in the proposed work.
11. *FogBus* In [31], the authors proposed a framework called FogBus to minimize the data traffic by minimizing the network and CPU usage in IoT-Fog-Cloud infrastructure. Their proposed approach used a blockchain-based technique to minimize the high data traffic and secure the private confidential IoT data from outside intruders and hackers. It applies several encryption techniques to secure operations on IoT sensitive data. Besides, the proposed framework facilitates the IoT-Fog (Edge)-Cloud infrastructure integration.
12. *Non-orthogonal multiple access* In [37], the authors applied a non-orthogonal multiple access technique to achieve ultra-low latency in IoT networks for short packets. The technique was able to minimize the queuing delay. The prosed work was optimal to be used for shot-packet communication in IoT networks. The method can be used for both uplink and downlink communication.
13. *Blockchain-based framework* In [78], the authors proposed a blockchain-based framework for IoT. The frameworks work in a decentralized manner to secure data transmission from IoT to the cloud. Their framework was able to minimize latency when compared with other existing permission less framework such as Ethereum.
14. *Job-scheduling algorithm* In [79], the authors proposed a novel job scheduling algorithm for the delay and performance optimization in the FC environment for IoT. They highlighted the issue of large data transmission in IoT which created a bottleneck over a network. This further lead to queueing delay and high network latency. Their proposed algorithm

works as an FC scheduler for application and service request in IoT networks. It optimizes delay and network usage and minimizes energy consumption. They have used the iFogSim simulator. The result generated shows that the novel algorithm was able to minimize delay and network usage by 32% and 16% when compares with other existing techniques.

RQ3 How the network of IoTs and cloud can meet all the QoS requirements for time-sensitive applications?

Few applications in the healthcare sector require the data in real-time. In such cases, FC-based technology is required to fulfil the QoS requirements of healthcare IoT. Using this technology, the IoT device can transfer the data in real-time. FC eases the communication between fog nodes and end devices by minimizing the high data traffic. These devices communicate using Bluetooth and Zigbee [61]. FC record PHD and vital signs at the edge of networks. These devices in healthcare are mostly used in remote places. When compares to the cloud, the FC meets the QoS requirements for healthcare time-sensitive applications. FC works in a distributed local area network (LAN). Whereas cloud works more in a centralized manner with a wide area network (WAN). FC device is an intelligent device. Many time-sensitive healthcare applications can be run in fog devices as it acts near to IoT devices [21]. The large geographical deployment of smart edge devices and applications that require real-time data processing, has with no doubt created the need to extend the reach of cloud computing to the edge, recently also referred to as fog or edge computing. FC is designed to complement cloud computing, paving the way for a novel, enriched architecture that can benefit from and includes both fog and cloud resources.

1. *FogPlan* To meet the QoS requirement for low latency the authors in [28], introduced a framework called FogPlan. The framework was based on dynamic service providing applications to fog nodes. They used a greedy algorithm in their proposed work. The proposed work lacks the protocol design for services used between fog nodes. There are also no learning methods for traffic control between the fog nodes.
2. *FogTorch* In [4], the authors proposed a model for latency and bandwidth minimization based on a Java tool called FogTorch in IoT and cloud. The model supports the QoS requirement of IoTs. A fog infrastructure was deployed by authors in their model, but they did not work on the reliability, scheduling of fog nodes deployment, and cost of the model.
3. *Analysis and comparison for e-healthcare requirements* Similarly, in [9], the authors discussed the QoS requirements for e-healthcare. This includes healthcare IoT and telemedicine requirements for real-time data transmission. The requirement for IoT devices and telemedicine operations is minimum round-trip time delay i.e. minimum service delay. The authors executed a detailed analysis and comparison of QoS requirements for e-healthcare services.
4. *FC and smart healthcare* In [80], the authors discussed the role of IoTs for smart healthcare applications. Their discussion includes various technologies, several challenges and opportunities associated with healthcare IoT, e-healthcare, and telemedicine. The authors further discussed the role of FC to minimize the high latency and meet QoS requirements for time-sensitive application in smart healthcare. They proposed a unique model to monitor patient health conditions. Next, the authors discussed the various wearable device to monitor and record patient vital signs. They used a machine learning approach in their proposed system.
5. *Healthfog* In [30], the authors proposed a framework for a smart healthcare system using deep learning-based techniques to diagnose the patient heart disease in real-time mode by integrating IoT and FC. The proposed framework was able to meet the QoS requirement for healthcare IoT. The framework was designed to operate for latency-sensitive applications like healthcare monitoring and flight control. Healthcare IoT generates a large amount of data called Big data. This data requires a large computation, next the data is transferred to databases and from databases to cloud data centres which lead to a drop in the performance of the system. Therefore, the proposed fog-enabled cloud framework meets the QoS for healthcare IoT in terms of power consumption, jitter, and prediction accuracy of heart disease diagnosis.
6. *Genetic metaheuristic algorithm* In [81], the authors developed a novel genetic metaheuristic algorithm with a QoS planner for network function virtualization in an IoT platform. Their prosed technique consists of server virtualization for latency minimization in IoT network. Furthermore, they have formulated a multi-objective optimization problem for scaling and deployment of IoT devices over a network to meet the requirement of minimum latency for time-sensitive applications.
7. *Data-oriented approach* In [82], the authors proposed a data-oriented approach to achieve ultra-low latency by meeting the criteria of reliability in wireless

communication. Their proposed approach meets the QoS requirement for IoT in terms of minimum latency.

8. *Smart architecture* In [83], the authors proposed architecture for smart manufacturing system for service latency in edge computing, fog computing, and cloud computing. Furthermore, the proposed novel system was able to handle large voluminous of data generated from IoT devices and meet QoS requirements.

5 Performance tools for IoT devices

This section discussed the available simulation tools used to evaluate the performance of the IoT devices. Furthermore, a detailed comparative analysis of existing tools is highlighted in this section.

Table 8 summarizes the available simulation tools. The first column refers to the problem studied by the various simulators. The second column lists the metrics reported by

Table 8 Comparison of modelling tools in IoT

Problem studied	Metrics	Simulation Tools
Source of latency in healthcare [30]	Communication latency	iFogSim
Optimal distribution of processing load among the fog and the cloud [21]	Processing cost per unit time	CloudSim
Quality of service (QoS) in IoT networks [4]	Service delay, propagation and transmission delays, processing delay	FogTorch: A Java Tool and iFogSim
Allocation of FC resources under QoS constraints [73]	CPU utilization, system response time, system loss rate and system throughput, number of messages	FogNetSim++
Modelling a typical healthcare monitoring system [31]	Computing cost and response time	FogBus
Resource allocation [4]	Price and time cost	HealthFog and FogTorch
Services migration from the edge to the cloud [24]	Operational cost, transmission cost, routing costs, reconfiguration cost	iFogSim
Offloading process optimization [20]	Processing and storage costs	ModFogSim and CloudSim
FC performance for IoT applications [24]	Latency and energy consumption	iFogSim
IoT resource management and FC performance [84]	Latency and bandwidth consumption	NetSim

Table 9 Simulator tools for IoT

Simulator	Implementation technologies	Metrics	Objective
Edge-Fog [77]	Python	QoS and energy consumption	Distribute task processing on the participating cloud resources
FogTorch [4]	Java	QoS, reliability of links and nodes, power consumption, security, monetary costs	Find eligible deployments of an application over a fog infrastructure
FogTorch II [28]	Extension of FogTorch	Resource utilization and QoS accuracy	Same as FogTorch and many QoS profile according to a probability distribution
iFogSim [30]	Extension of CloudSim, Java, JSON	Energy consumption, network congestion, and operational costs	Performance of resource management policies
MyiFogSim [20]	Extension of iFogSim	Latency	Resource allocation
FogNetSim [73]	Based on OMNeT++	Energy module, scheduling algorithms, pricing model	General simulation of fog environments
Fogbus [31]	Use Java and iFogSim	Latency, energy, network and CPU usage	Resource management
FogTorch [4]	Based on CloudSim	Computational and network cost	IoT resource management in FC
OMNeT++ [81]	Component-based C++ simulation	Scheduling and latency	IoT resource management

Table 10 Challenges (Ci, i = 1,2, 3…n)

C1: High network latency	[38, 41, 48, 60, 73, 74, 85–90]
C2: High communication latency	[14, 24, 28, 43, 57, 61, 67, 75, 91–95]
C3: High computation latency	[4, 24, 29, 40, 42, 62, 64, 65, 77, 96–98]
C4: High data traffic	[38, 48, 56, 59, 67, 85, 88, 92, 97–101]
C5: High bandwidth consumption	[4, 57, 66, 88–90, 102]
C6: Large volume of data	[66, 85, 95, 97–99]
C7: Real -time data transmission	[87, 91, 103–108]
C8: Load balancing and data replication	[4, 41, 58, 59, 88, 102, 109]
C9: High Energy consumption	[20, 63, 100, 110]
C10: Coordination between IoT, Fog and Cloud	[64, 66, 96, 111]

Table 11 Advantages (Adi,, i = 1,2, 3…. N)

AD1: Minimized Network Latency	[38, 40, 41, 56, 60, 78, 85, 87, 90, 94, 97]
AD2: Minimized data traffic	[56, 57, 65, 93, 94, 96, 99]
AD3: Minimized response time	[28, 73, 86, 103, 105]
AD4: Minimized Network cost	[38, 63, 67, 107]
AD5: Minimized communication latency	[4, 14, 20, 28, 43, 48, 61, 75, 77, 87, 91, 92]
AD6: Minimized Network usage	[59, 88, 93, 97–99, 111]
AD7: Load balance	[58, 62, 66, 88, 93, 102]
AD8: Reduced packet loss	[28, 89, 99, 104, 106]
AD9: Real-time data transmission	[24, 91, 98, 100, 101, 104]
AD10: Reduced energy consumption	[20, 40, 88, 90]
AD11: Highly distributed, computation and storage	[64, 66, 74, 111]
AD12: Minimized bandwidth	[4]
AD13: Minimized computation latency	[24, 29, 41, 42, 62, 65, 77]

Table 12 Proposed solution (Si, i = 1,2, 3…. N) for the challenges (C_i, i = 1,2, 3…n) mentioned in Table 10

S1: FC architecture	[40, 57, 62, 85]
S2: Smart gateway using FC	[24, 87, 91, 98, 99]
S3: Cloud-based healthcare system	[103]
S4: FC-based smart health gateway	[87, 97]
S5: Resource aware placement	[59, 64, 66, 86, 112]
S6: Efficient load balancing using FC	[54, 58, 59, 88, 93, 102]
S7: Network model	[60, 64, 104]
S8: Multi-Tier based cloud system	[105]
S9: Cloud-based WBAN	[100]
S10: Green cloud-assisted healthcare service on WBAN	[106]
S11: FC merged with VANETs	[92]
S12: FogPlan	[28]
S13: FogTorch	[4]
S14: F-RAN	[29]
S15: Fog-to-fog communication	[75]
S16: Live migration using FC	[43]
S17: Computation offloading solution	[20, 96]
S18: Software Defined Network (SDN)	[48, 94, 107]
S19: A fog scheme-based algorithm	[73]
S20: Cloudlet based solution	[14]
S21: Reinforcement learning-based load balancing	[88]

Table 12 (continued)

S22: Virtual Fog	[63, 77]
S23: Layered F2C architecture	[111]
S24: Integration of cloud platform and IoT	[101]
S25: Open-loop communication for the edge devices	[38, 67]
S26: Design and develop a concept called smart data taking the advantage of FC	[89]
S27: Hybrid platform as a service designed for IoT	[90]
S28: NB-IoT system	[41]
S29: Surface-aided MEC	[42, 65]
S30: Fuzzy-based FC system	[61]

Table 13 Research Gaps (RGi, i = 1,2, 3……n)

RG1: No work has been done on minimizing service and computation latency	[40, 58, 85, 95]
RG2: No work has been done to minimize high latency	[66, 99]
RG3: No work has been to minimize high computation latency	[56, 86, 96]
RG4: Proposed system lacks work on computation and network latency	[61, 63, 103]
RG5: No work has been done on computation latency and RAM consumption	[66, 87]
RG6: Research work not focused on network traffic reduction and no work has been done on latency minimization	[57, 111]
RG7: No work has been done to minimize high service latency	[65, 102]
RG8: No work has been done to minimize high computation, network latency, and network usage	[91]
RG9: No work on computation latency	[59, 60, 104]
RG10: No work has been done on network traffic	[41, 62, 105]
RG11: No work has been done on RAM consumption and computation latency	[100]
RG12: No work has been done on minimizing network latency and computational latency	[106]
RG13: Research work lacks the minimization of computation and network latency	[101]
RG14: Not considered the latency due to multiple hop count	[92]
RG15: Not considered service latency	[38]
RG16: Not considered communication and computation delay	[89]
RG17: Not considered RAM consumption, communication, and computation latency between IoTs and cloud	[90]
RG18: VM Migration and fog services discovery issues are not considered	[28, 64]
RG19: The reliability of fog node deployment is missing. Deployment cost is high	[4]
RG20: The proposed model is not validated for real-life implementations. No work has been done on the task processing history of fog nodes	[75]
RG21: No discussion on the high network latency between wireless devices and fog nodes	[29]
RG22: No real-world implementation. Future work will include deploying and validation the framework	[43]
RG23: Not discussed the network latency due to large data transmission and no real-world implementation	[24]
RG24: No work has been done on high latency generated due to large data transmission	[20, 67]
RG25: Not able to perform energy-efficient communication	[42, 107]
RG26: Not considered the computation time	[73]
RG27: Network traffic is not considered for implementation	[14]
RG28: Model-free learning is required to merge with model-based learning	[88]
RG29: Did not discuss the network latency	[93]
RG30: Research works lack continuous communication between the fog nodes	[48]
RG31: Problem with heterogeneous devices, synchronization and scheduling	[77]
RG32: Unable to exchange multicast data to different nodes	[94]
RG33: Data Security, network usage and RAM consumption is not considered	[96]

Table 14 Comparative analysis of the existing works for latency minimization

Authors and year	Challenges	Solutions	Advantages	Research gaps
1. W. Lee et al. [85], 2016	C1 C4 C6	S1	AD1	RG1
2. M. Aazam et al. [99], 2014	C4 C6	S2	AD2 AD6 AD8	RG2
3. Y. Shi et al. [97], 2015	C3 C4 C6	S4	AD1 AD6	RG1
4. M. Taneja et al. [86], 2016	C1	S5	AD3	RG3
5. K. Sundharakumar et al. [103], 2015	C7	S3	AD3	RG4
6.A.M.Rahmani et al. [87], 2017	C1 C7	S2 S4	AD1	RG5
7. T. N. Gia et al. [98], 2015	C3 C4 C6	S2	AD6	RG3
8. X. Masip-Bruin et al. [111], 2016	C10	S23	AD11	RG6
9. S. Verma et al. [102], 2016	C8	S6	AD7	RG7
10. P. Marie et al. [91], 2016	C7	S2	AD5 AD9	RG8
11. M. M. Hassan et al. [104], 2017	C7	S7	AD8	RG9
12.M. Quwaider et al. [105], 2016	C7	S8	AD3	RG10
13. O. Diallo et al. [100], 2013	C4	S9	AD9	RG11
14. H.-P. Chiang et al. [106], 2014	C7	S10	AD8	RG12
15. F. de Arriba-Pérez et al. [101], 2016	C4	S24	AD9	RG13
16. K. Kai et al. [92], 2016	C4	S11	AD5	RG14
17. S.-C. Hung et al. [38], 2015	C1	S25	AD4	RG15
18. F. Hosseinpour et al. [67], 2016	C1	S26	AD8	RG16
19. O. Bibani et al. [90], 2016	C1	S27	AD1	RG17
20. A. Yousefpour et al. [28], 2019	C2	S12	AD3 AD5	RG18
21. A. Brogi et al. [4], 2017	C3 C5	S13	AD1 AD12	RG19
22. W. Masri et al. [40], 2017	C2	S15	AD5	RG1
23. A.-C. Pang et al. [34], 2017	C3	S14	AD13	RG21
24. O. Osanaiye et al. [43], 2017	C2	S16	AD5	RG22
25. F. A. Kraemer et al. [25], 2019	C2	S2	AD9 AD13	RG23
26. X. Meng et al. [21], 2017	C9	S17	AD10	RG24
27. P. K. Sharma et al. [69], 2017	C7	S18	AD4	RG25
28. H. A. M. Name et al. [73], 2017	C1	S19	AD3	RG26
29. M. Ali et al. [14], 2018	C2	S20	AD5	RG27
30. J.-y. Baek et al. [88], 2019	C1 C4	S21	AD10 AD6	RG28
31. R. Deng et al. [93], 2016	C2	S6	AD2 AD7	RG29
32. J. Wang et al. [48], 2018	C1	S18	AD5	RG30
33. J. Li et al. [77], 2017	C3	S22	AD13	RG31
34. L. Gao et al. [94], 2016	C2	S18	AD1 AD2	RG32
35. L. Lyu et al. [96], 2017	C10	S17	AD2	RG33
36. L. Shi et al. [33], 2020	C3	S2	AD1	RG3
37.A.K. Sultania et al. [34], 2020	C1	S28	AD1	RG11
38. T. Bai et al. [35], 2020	C3	S29	AD13	RG25
39. F. Banaie et al. [49], 2020	C4	S6	AD1AD2	RG3
40.I.Martinez et al. [50], 2020	C2	S1	AD2	RG6
41.M. Goudarzi et al. [51], 2020	C8	S6	AD7	RG1
42. Z. Chang et al. [52], 2020	C4C8	S6	AD6	RG9
43.R.O. Aburukba et al. [58], 2020	C3	S29	AD13	RG7
44. S. K. Singh et al. [59], 2020	C5C6	S5	AD7AD11	RG2
45. H. Lu et al. [60], 2020	C2	S25	AD4	RG24

Table 14 (continued)

Authors and year	Challenges	Solutions	Advantages	Research gaps
46. C. Chen et al. [61], 2020	C1	S7	AD1	RG9
47. T. T. Khanh et al. [62], 2020	C2	S30	AD5	RG4
48.A.Gowri et al. [63], 2020	C3	S1	AD13	RG10
49.P. Gazori et al. [64], 2020	C9	S22	AD4	RG4
50. J. Jiang et al. [65], 2020	C3C10	S5S7	AD11	RG18
51. H. Ren et al. [67], 2020	C10	S5	AD11	RG5
52. O. Elgarhy et al. [68], 2020	C8	S8	AD13	RG7
53.M. Mudassar et al. [69], 2020	C10	S23	AD7	RG9
54. C. Tian et al. [70], 2020	C6	S21	AD10	RG15
55.P. Brous et al. [84], 2020	C4	S19	AD5	RG21

each simulator which further highlights the main objectives of each simulator. The third column refers to the list of simulators (Tables 9, 10, 11, 12, 13, 14).

6 Identified research gaps and comparative analysis

The recent research work lacks the suggestion related to minimizing the latency. The advancement in the field of IoT requirements and challenges associated with it has brought many research and development issues. There is a lot of enthusiasm and interest in the field of IoT applications. However, very little empirical study has been done in the field of QoS requirement for time-sensitive applications in IoT. need

7 Conclusion

The paper discusses the recent latency techniques and technologies in the cloud and IoT. The paper uses a PRISMA technique to identify the research challenges and gaps in the field of IoT. Furthermore, we identified, and cluster 23 approaches and 32 technologies associated with latencies in the cloud and IoT. A total of 112 papers on latency reduction have been examined. Moreover, this study has systematically identified and rigorously reviewed relevant papers and synthesized the data extracted from those papers to answers a set of research questions that motivated this review. The goal of this SLR is to give an inside observation of the latency issue in IoTs and cloud and to propose a new solution and comparison of the other known latency reduction techniques. The aim is to motivate future works to develop a latency aware model for IoT, cloud, and end-users. From this survey, the researchers and industrialists will be able to gain an insight view of IoT and cloud requirements for time-sensitive applications with a betterarxiv understanding of latency reduction techniques and technologies.

Acknowledgements The author would like to thank the Centre of Graduate Studies, Computer and Information Science Department, Universiti Teknologi PETRONAS, Malaysia for their expertise and cooperation in this research.

Author contributions SS, MFH, DCT: Conceptualization. SS, MFH, DCT: Formal analysis. SS, MFH, IVP: Investigation. MFH, MKK: Supervision. SS: Writing-original draft. SS, MFH, DCT, RA: Writing-review & editing.

Funding Using project allowance and self-finance.

Data availability Not required in the review article.

Declarations

Conflict of interest Saurabh Shukla, Mohd. Fadzil Hassan, Duc Chung Tran, Rehan Akbar, Irving Vitra Paputungan and Muhammad Khalid Khan declare that they have no competing interests.

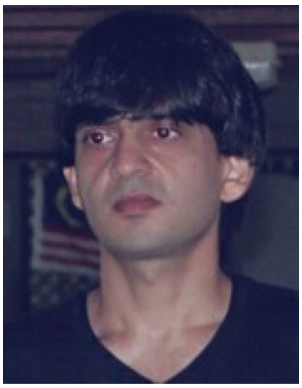
References

1. Hammi, B., Khatoun, R., Zeadally, S., Fayad, A., Khoukhi, L.: IoT technologies for smart cities. IET Netw. **7**(1), 1–13 (2017)
2. Wortmann, F., Flüchter, K.: Internet of things. Bus. Inf. Syst. Eng. **57**(3), 221–224 (2015)
3. Shukla, S., Hassan, M.F., Khan, M.K., Jung, L.T., Awang, A.: An analytical model to minimize the latency in healthcare internet-of-things in fog computing environment. PLoS ONE **14**(11), e0224934 (2019)
4. Brogi, A., Forti, S.: QoS-aware deployment of IoT applications through the fog. IEEE Internet Things J. **4**(5), 1185–1192 (2017)
5. Alicherry, M., Lakshman, T.: Optimizing data access latencies in cloud systems by intelligent virtual machine placement. In: 2013 Proceedings IEEE INFOCOM. IEEE, pp. 647–655 (2013)
6. Pan, J., McElhannon, J.: Future edge cloud and edge computing for internet of things applications. IEEE Internet Things J. **5**(1), 439–449 (2017)

7. Nandyala, C.S., Kim, H.-K.: From cloud to fog and IoT-based real-time U-healthcare monitoring for smart homes and hospitals. Int. J. Smart Home **10**(2), 187–196 (2016)
8. Sun, X., Ansari, N.: Latency aware workload offloading in the cloudlet network. IEEE Commun. Lett. **21**(7), 1481–1484 (2017)
9. Skorin-Kapov, L., Matijasevic, M.: Analysis of QoS requirements for e-health services and mapping to evolved packet system QoS classes. Int. J. Telemed. Appl. **2010**, 9 (2010)
10. Alam, M.G.R., Tun, Y.K., Hong, C.S.: Multi-agent and reinforcement learning based code offloading in mobile fog. In: 2016 International Conference on Information Networking (ICOIN). IEEE, pp. 285–290 (2016)
11. Kao, Y.-H., Krishnamachari, B., Ra, M.-R., Bai, F.: Hermes: latency optimal task assignment for resource-constrained mobile computing. IEEE Trans. Mob. Comput. **16**(11), 3056–3069 (2017)
12. Nishtala, R., Carpenter, P., Petrucci, V., Martorell, X.: Hipster: Hybrid task manager for latency-critical cloud workloads. In: 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE, pp. 409–420 (2017)
13. Sajithabanu, S., Balasundaram, S.: Cloud based Content Delivery Network using Genetic Optimization Algorithm for storage cost. In: 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). IEEE, pp. 1–6 (2016)
14. Ali, M., Riaz, N., Ashraf, M.I., Qaisar, S., Naeem, M.: Joint cloudlet selection and latency minimization in fog networks. IEEE Trans. Ind. Inf. **14**(9), 4055–4063 (2018)
15. Naas, M.I., Parvedy, P.R., Boukhobza, J., Lemarchand, L.: iFogStor: an IoT data placement strategy for fog infrastructure. In: 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC). IEEE, pp. 97–104 (2017)
16. Grinnemo, K.-J., Brunstrom, A.: A first study on using MPTCP to reduce latency for cloud based mobile applications. In: 2015 IEEE Symposium on Computers and Communication (ISCC). IEEE, pp. 64–69 (2015)
17. Habak, K., Ammar, M., Harras, K.A., Zegura, E.: Femto clouds: Leveraging mobile devices to provide cloud service at the edge. In: 2015 IEEE 8th international conference on cloud computing. IEEE, pp. 9–16 (2015)
18. Lee, M., Kim, Y., Lee, Y.: A home cloud-based home network auto-configuration using SDN. In: 2015 IEEE 12th International conference on networking, sensing and control. IEEE, pp. 444–449 (2015)
19. Bi, Y., Han, G., Lin, C., Deng, Q., Guo, L., Li, F.: Mobility support for fog computing: an SDN approach. IEEE Commun. Mag. **56**(5), 53–59 (2018)
20. Meng, X., Wang, W., Zhang, Z.: Delay-constrained hybrid computation offloading with cloud and fog computing. IEEE Access **5**, 21355–21367 (2017)
21. Cao, H., Cai, J.: Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: a game-theoretic machine learning approach. IEEE Trans. Veh. Technol. **67**(1), 752–764 (2017)
22. Kargatzis, D., Sotiriadis, S., Petrakis, E.G.: Virtual machine migration in heterogeneous clouds: from openstack to VMWare. In: 2017 IEEE 38th Sarnoff Symposium. IEEE, pp. 1–6 (2017)
23. Eccles, M.J., Evans, D.J., Beaumont, A.J.: True real-time change data capture with web service database encapsulation. In: 2010 6th World Congress on Services. IEEE, pp. 128–131 (2010)
24. Kraemer, F.A., Braten, A.E., Tamkittikhun, N., Palma, D.: Fog computing in healthcare—a review and discussion. IEEE Access **5**, 9206–9222 (2017)
25. Sambyo, K., Bhunia, C.T.: Application of multi level ATM in reducing latency in clouds for performance improvement of integrated voice, video and data services. In: 2014 11th International Conference on Information Technology: New Generations. IEEE, pp. 607–607 (2014)
26. Qin, H.: Locpart: a latency optimized cache partitioning for cloud data centers. In: 2017 4th International Conference on Information Science and Control Engineering (ICISCE). IEEE, pp. 433–437 (2017)
27. Cho, D., Taheri, J., Zomaya, A.Y., Bouvry, P.: Real-time virtual network function (VNF) migration toward low network latency in cloud environments. In: 2017 IEEE 10th International Conference on Cloud Computing (CLOUD). IEEE, pp. 798–801 (2017)
28. Yousefpour, A. et al.: FogPlan: a lightweight QoS-aware dynamic fog service provisioning framework. IEEE Internet Things J. (2019)
29. Pang, A.-C., Chung, W.-H., Chiu, T.-C., Zhang, J.: Latency-driven cooperative task computing in multi-user fog-radio access networks. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). IEEE, pp. 615–624 (2017)
30. Tuli, S., et al.: Healthfog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated IoT and FoG computing environments. Futur. Gener. Comput. Syst. **104**, 187–200 (2020)
31. Tuli, S., Mahmud, R., Tuli, S., Buyya, R.: Fogbus: a blockchain-based lightweight framework for edge and fog computing. J. Syst. Softw. (2019)
32. Yang, H., Alphones, A., Zhong, W.-D., Chen, C., Xie, X.: Learning-based energy-efficient resource management by heterogeneous RF/VLC for ultra-reliable low-latency industrial IoT networks. IEEE Trans. Ind. Inf. **16**(8), 5565–5576 (2019)
33. Shi, L., Ahmad, I., He, Y., Chang, K.: Service group based FOFDM-IDMA platform to support massive connectivity and low latency simultaneously in the uplink IoT environment. Wirel. Commun. Mob. Comput. (2020)
34. Sultania, A.K., Mahfoudhi, F., Famaey, J.: Real-time demand-response using NB-IoT. IEEE Internet Things J. (2020)
35. Bai, T., Pan, C., Deng, Y., Elkashlan, M., Nallanathan, A., Hanzo, L.: Latency minimization for intelligent reflecting surface aided mobile edge computing. IEEE J. Sel. Areas Commun. **38**(11), 2666–2682 (2020)
36. Fent, P., van Renen, A., Kipf, A., Leis, V., Neumann, T., Kemper, A.: Low-latency communication for fast DBMS using RDMA and shared memory. In: 2020 IEEE 36th International Conference on Data Engineering (ICDE). IEEE, pp. 1477–1488 (2020)
37. Xiang, Z., Yang, W., Cai, Y., Ding, Z., Song, Y., Zou, Y.: NOMA-assisted secure short-packet communications in IoT. IEEE Wirel. Commun. **27**(4), 8–15 (2020)
38. Hung, S.-C., Liau, D., Lien, S.-Y., Chen, K.-C.: Low latency communication for Internet of Things. In: 2015 IEEE/CIC International Conference on Communications in China (ICCC). IEEE, pp. 1–6 (2015)
39. Yousefpour, A., Ishigaki, G., Jue, J.P.: Fog computing: towards minimizing delay in the internet of things. In: 2017 IEEE international conference on edge computing (EDGE). IEEE, pp. 17–24 (2017)
40. Mukherjee, M., Shu, L., Wang, D.: Survey of fog computing: fundamental, network applications, and research challenges. IEEE Commun. Surv. Tutor. **20**(3), 1826–1857 (2018)
41. Mouradian, C., Naboulsi, D., Yangui, S., Glitho, R.H., Morrow, M.J., Polakos, P.A.: A comprehensive survey on fog computing: state-of-the-art and research challenges. IEEE Commun. Surv. Tutor. **20**(1), 416–464 (2017)

42. Bittencourt, L. et al.: The internet of things, fog and cloud continuum: integration and challenges. Internet of Things (2018)
43. Osanaiye, O., Chen, S., Yan, Z., Lu, R., Choo, K.-K.R., Dlodlo, M.: From cloud to fog computing: a review and a conceptual live VM migration framework. IEEE Access **5**, 8284–8300 (2017)
44. Wu, J., Dong, M., Ota, K., Li, J., Guan, Z.: FCSS: Fog computing based content-aware filtering for security services in information centric social networks. IEEE Trans. Emerg. Topics Comput. (2017)
45. Dinh, N.-T., Kim, Y.: An efficient availability guaranteed deployment scheme for IoT service chains over Fog-Core Cloud Networks. Sensors **18**(11), 3970 (2018)
46. Li, G., Wu, J., Li, J., Wang, K., Ye, T.: Service popularity-based smart resources partitioning for fog computing-enabled industrial Internet of Things. IEEE Trans. Industr. Inf. **14**(10), 4702–4711 (2018)
47. Mahmud, R., Koch, F.L., Buyya, R.: Cloud-fog interoperability in IoT-enabled healthcare solutions. In: Proceedings of the 19th International Conference on Distributed Computing and Networking. ACM, p. 32 (2018)
48. Wang, J., Li, D.: Adaptive computing optimization in software-defined network-based industrial Internet of Things with Fog Computing. Sensors **18**(8), 2509 (2018)
49. Banaie, F., Yaghmaee, M.H., Hosseini, A., Tashtarian, F.: Load-balancing algorithm for multiple gateways in Fog-based Internet of Things. IEEE Internet Things J. (2020)
50. Martinez, I., Jarray, A., Hafid, A.S.: Scalable design and dimensioning of Fog-Computing infrastructure to support latency sensitive IoT applications. IEEE Internet Things J. (2020)
51. Goudarzi, M., Wu, H., Palaniswami, M.S., Buyya, R.: An application placement technique for concurrent iot applications in edge and fog computing environments. IEEE Trans. Mob. Comput. (2020)
52. Chang, Z., Liu, L., Guo, X., Sheng, Q.: Dynamic resource allocation and computation offloading for IoT Fog computing system. IEEE Trans. Ind. Inform. (2020)
53. Awaisi, K.S., Hussain, S., Ahmed, M., Khan, A.A., Ahmed, G.: Leveraging IoT and Fog computing in healthcare systems. IEEE Internet Things Mag. **3**(2), 52–56 (2020)
54. Soo, S., Chang, C., Loke, S.W., Srirama, S.N.: Dynamic Fog Computing: practical processing at mobile edge devices. In: Algorithms, Methods, and Applications in Mobile Computing and Communications: IGI Global, pp. 24–47 (2019)
55. Bellavista, P., Berrocal, J., Corradi, A., Das, S.K., Foschini, L., Zanni, A.: A survey on fog computing for the Internet of Things. Pervasive Mob. Comput. **52**, 71–99 (2019)
56. Soleymani, S.A., et al.: A secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing. IEEE Access **5**, 15619–15629 (2017)
57. Rafique, H., Shah, M.A., Islam, S.U., Maqsood, T., Khan, S., Maple, C.: A novel bio-inspired hybrid algorithm (NBIHA) for efficient resource management in fog computing. IEEE Access **7**, 115760–115773 (2019)
58. Aburukba, R.O., AliKarrar, M., Landolsi, T., El-Fakih, K.: Scheduling Internet of Things requests to minimize latency in hybrid Fog–Cloud computing. Futur. Gener. Comput. Syst. **111**, 539–551 (2020)
59. Singh, S.K., Rathore, S., Park, J.H.: Blockiotintelligence: a blockchain-enabled intelligent IoT architecture with artificial intelligence. Futur. Gener. Comput. Syst. **110**, 721–743 (2020)
60. Lu, H., He, X., Du, M., Ruan, X., Sun, Y., Wang, K.: Edge QoE: computation offloading with deep reinforcement learning for Internet of Things. IEEE Internet Things J. (2020)
61. Chen, C., Chen, Y., Zhang, K., Ni, M., Wang, S., Liang, R.: System redundancy enhancement of secondary frequency control under latency attacks. IEEE Trans. Smart Grid (2020)
62. Khanh, T.T., Oo, T.Z., Tran, N.H., Huh, E.-N., Hong, C.S.: Latency minimization in a fuzzy-based mobile edge orchestrator for IoT applications. IEEE Commun. Lett. (2020)
63. Gowri, A.: Fog resource allocation through machine learning algorithm. In: Architecture and Security Issues in Fog Computing Applications: IGI Global, pp. 1–41 (2020)
64. Gazori, P., Rahbari, D., Nickray, M.: Saving time and cost on the scheduling of fog-based IoT applications using deep reinforcement learning approach. Futur. Gener. Comput. Syst. **110**, 1098–1115 (2020)
65. Jiang, J., Li, Z., Tian, Y., Al-Nabhan, N.: A review of techniques and methods for IoT applications in collaborative Cloud-Fog Environment. Secur. Commun. Netw. (2020)
66. Zhang, L., Ansari, N.: Latency-aware IoT service provisioning in UAV-aided mobile-edge computing networks. IEEE Internet Things J. **7**(10), 10573–10580 (2020)
67. Ren, H., Pan, C., Deng, Y., Elkashlan, M., Nallanathan, A.: Resource allocation for secure URLLC in mission-critical IoT scenarios. IEEE Trans. Commun. **68**(9), 5793–5807 (2020)
68. Elgarhy, O., Reggiani, L., Malik, H., Alam, M.M., Imran, M.A.: Rate-latency optimization for NB-IoT with adaptive resource unit configuration in uplink transmission. IEEE Syst. J. (2020)
69. Mudassar, M., Zhai, Y., Liao, L., Shen, J.: A decentralized latency-aware task allocation and group formation approach with fault tolerance for IoT applications. IEEE Access **8**, 49212–49223 (2020)
70. Tian, C., et al.: P-PFC: reducing tail latency with predictive PFC in lossless data center networks. IEEE Trans. Parallel Distrib. Syst. **31**(6), 1447–1459 (2020)
71. Cavalcante, E., et al.: On the interplay of Internet of Things and Cloud Computing: a systematic mapping study. Comput. Commun. **89**, 17–33 (2016)
72. Liu, Y., Fieldsend, J.E., Min, G.: A framework of fog computing: Architecture, challenges, and optimization. IEEE Access **5**, 25445–25454 (2017)
73. Name, H.A.M., Oladipo, F.O., Ariwa, E.: User mobility and resource scheduling and management in fog computing to support IoT devices. In: 2017 Seventh International Conference on Innovative Computing Technology (INTECH). IEEE, pp. 191–196 (2017)
74. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M.: Internet of things: a survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutor. **17**(4), 2347–2376 (2015)
75. Masri, W., Al Ridhawi, I., Mostafa, N., Pourghomi, P.: Minimizing delay in IoT systems through collaborative fog-to-fog (F2F) communication. In: 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN). IEEE, pp. 1005–1010 (2017)
76. Naha, R.K., et al.: Fog Computing: survey of trends, architectures, requirements, and research directions. IEEE access **6**, 47980–48009 (2018)
77. Li, J., Jin, J., Yuan, D., Zhang, H.: Virtual fog: a virtualization enabled fog computing framework for Internet of Things. IEEE Internet Things J. **5**(1), 121–131 (2017)
78. Seshadri, S.S. et al.: Iotcop: a blockchain-based monitoring framework for detection and isolation of malicious devices in internet-of-things systems. IEEE Internet Things J. (2020)
79. Jamil, B., Shojafar, M., Ahmed, I., Ullah, A., Munir, K., Ijaz, H.: A job scheduling algorithm for delay and performance optimization in fog computing. Concurr. Comput. **32**(7), e5581 (2020)

80. Baker, S.B., Xiang, W., Atkinson, I.: Internet of things for smart healthcare: technologies, challenges, and opportunities. IEEE Access **5**, 26521–26544 (2017)
81. Ouedraogo, C.A., Medjiah, S., Chassot, C., Drira, K., Aguilar, J.: A cost-effective approach for end-to-end QoS management in NFV-enabled IoT platforms. IEEE Internet Things J. (2020)
82. Yang, H.-C., Bao, T., Alouini, M.-S.: Transient performance limits for ultra-reliable low-latency communications over fading channels. IEEE Trans. Veh. Technol. **69**(11), 13970–13973 (2020)
83. Qi, Q., Tao, F.: A smart manufacturing service system based on edge computing, fog computing, and cloud computing. IEEE Access **7**, 86769–86777 (2019)
84. Brous, P., Janssen, M., Herder, P.: The dual effects of the Internet of Things (IoT): a systematic review of the benefits and risks of IoT adoption by organizations. Int. J. Inf. Manag. **51**, 101952 (2020)
85. Lee, W., Nam, K., Roh, H.-G., Kim, S.-H.: A gateway based fog computing architecture for wireless sensors and actuator networks. In: 2016 18th International Conference on Advanced Communication Technology (ICACT). IEEE, pp. 210–213 (2016)
86. Taneja, M., Davy, A.: Resource aware placement of data analytics platform in fog computing. Procedia Comput. Sci. **97**, 153–156 (2016)
87. Rahmani, A.M., et al.: Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: a fog computing approach. Futur. Gener. Comput. Syst. **78**, 641–658 (2018)
88. Baek, J.-Y., Kaddoum, G., Garg, S., Kaur, K., Gravel, V.: Managing Fog Networks using reinforcement learning based load balancing algorithm. arXiv preprint arXiv:1901.10023 (2019)
89. Hosseinpour, F., Plosila, J., Tenhunen, H.: An approach for smart management of big data in the fog computing context. In: 2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom). IEEE, pp. 468–471 (2016)
90. Bibani, O. et al.: A demo of iot healthcare application provisioning in hybrid cloud/fog environment. In: 2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom). IEEE, pp. 472–475 (2016)
91. Marie, P., Desprats, T., Chabridon, S., Sibilla, M.: Enabling self-configuration of QoC-centric fog computing entities. In: 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld). IEEE, pp. 526–533 (2016)
92. Kai, K., Cong, W., Tao, L.: Fog computing for vehicular ad-hoc networks: paradigms, scenarios, and issues. J. China Univ. Posts Telecommun. **23**(2), 56–96 (2016)
93. Deng, R., Lu, R., Lai, C., Luan, T.H., Liang, H.: Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption. IEEE Internet Things J. **3**(6), 1171–1181 (2016)
94. Gao, L., Luan, T.H., Yu, S., Zhou, W., Liu, B.: FogRoute: DTN-based data dissemination model in fog computing. IEEE Internet Things J. **4**(1), 225–235 (2016)
95. Badawy, M.M., Ali, Z.H., Ali, H.A.: Qos provisioning framework for service-oriented internet of things (iot). Clust. Comput. pp. 1–17 (2019)
96. Lyu, L., Jin, J., Rajasegarar, S., He, X., Palaniswami, M.: Fog-empowered anomaly detection in IoT using hyperellipsoidal clustering. IEEE Internet Things J. **4**(5), 1174–1184 (2017)
97. Shi, Y., Ding, G., Wang, H., Roman, H.E., Lu, S.: The fog computing service for healthcare. In: 2015 2nd International Symposium on Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech). IEEE, pp. 1–5 (2015)
98. Gia, T.N., Jiang, M., Rahmani, A.-M., Westerlund, T., Liljeberg, P., Tenhunen, H.: Fog computing in healthcare internet of things: a case study on ECG feature extraction. In: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. IEEE, pp. 356–363 (2015)
99. Aazam, M., Huh, E.-N.: Fog computing and smart gateway based communication for cloud of things. In: 2014 International Conference on Future Internet of Things and Cloud. IEEE, pp. 464–470 (2014)
100. Diallo, O., Rodrigues, J.J., Sene, M., Niu, J.: Real-time query processing optimization for cloud-based wireless body area networks. Inf. Sci. **284**, 84–94 (2014)
101. de Arriba-Pérez, F., Caeiro-Rodríguez, M., Santos-Gago, J.: Collection and processing of data from wrist wearable devices in heterogeneous and multiple-user scenarios. Sensors **16**(9), 1538 (2016)
102. Verma, S., Yadav, A.K., Motwani, D., Raw, R., Singh, H.K.: An efficient data replication and load balancing technique for fog computing environment. In: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). IEEE, pp. 2888–2895 (2016)
103. Sundharakumar, K., Dhivya, S., Mohanavalli, S., Chander, R.V.: Cloud based fuzzy healthcare system. Procedia Comput. Sci. **50**, 143–148 (2015)
104. Hassan, M.M., Lin, K., Yue, X., Wan, J.: A multimedia healthcare data sharing approach through cloud-based body area network. Futur. Gener. Comput. Syst. **66**, 48–58 (2017)
105. Quwaider, M., Jararweh, Y.: Multi-tier cloud infrastructure support for reliable global health awareness system. Simul. Model. Pract. Theory **67**, 44–58 (2016)
106. Chiang, H.-P., Lai, C.-F., Huang, Y.-M.: A green cloud-assisted health monitoring service on wireless body area networks. Inf. Sci. **284**, 118–129 (2014)
107. Sharma, P.K., Chen, M.-Y., Park, J.H.: A software defined fog node based distributed blockchain cloud architecture for IoT. IEEE Access **6**, 115–124 (2017)
108. Diogo, P., Lopes, N.V., Reis, L.P.: An ideal IoT solution for real-time web monitoring. Clust. Comput. **20**(3), 2193–2209 (2017)
109. Pourghebleh, B., Hayyolalam, V.: A comprehensive and systematic review of the load balancing mechanisms in the Internet of Things. Clust. Comput. 1–21 (2019)
110. Patel, Y.S., Reddy, M., Misra, R.: Energy and cost trade-off for computational tasks offloading in mobile multi-tenant clouds. Clust. Comput. 1–32 (2021)
111. Masip-Bruin, X., Marín-Tordera, E., Tashakor, G., Jukan, A., Ren, G.-J.: Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems. IEEE Wirel. Commun. **23**(5), 120–128 (2016)
112. Ghanbari, Z., Navimipour, N.J., Hosseinzadeh, M., Darwesh, A.: Resource allocation mechanisms and approaches on the Internet of Things. Clust. Comput. **22**(4), 1253–1282 (2019)



Saurabh Shukla Postdoctoral researcher, Electrical and Electronics Department, Data Science Institute (DSI), National University of Ireland Galway (NUIG), Galway, Ireland.

Rehan Akbar (Assistant Professor) Ph.D. (Computer Science) Faculty of Information and Communication Technology, University Tunku Abdul Rahman, Malaysia.

Mohd. Fadzil Hassan Associate Professor, Computer and Information Science Ph.D. (Computer Science), Department, Universiti Teknologi PETRONAS, Perak, Malaysia.

Irving Vitra Paputungan (Assistant Professor) Ph.D. (Computer Science) Informatics Department, Universitas Islam Indonesia, Yogyakarta, Indonesia.

Duc Chung Tran (Assistant Professor) Ph.D. (Computer Science), Computing Fundamental Department, FPT University, Veitnam.

Muhammad Khalid Khan (Associate Professor) Ph.D. (Computer Science) College of Computing and Information Sciences, PAF Karachi Institute of Economics and Technology, Karachi, Pakistan.

Authors and Affiliations

Saurabh Shukla[1] · Mohd. Fadzil Hassan[2] · Duc Chung Tran[3] · Rehan Akbar[4] · Irving Vitra Paputungan[5] · Muhammad Khalid Khan[6]

✉ Saurabh Shukla
saurabhshkl.shukla@gmail.com;
saurabh.shukla@nuigalway.ie

Mohd. Fadzil Hassan
mfadzil_hassan@utp.edu.my

Rehan Akbar
rehan@utar.edu.my

[1] Electrical and Electronics Department, Data Science Institute (DSI), National University of Ireland Galway (NUIG), Galway, Ireland

[2] Department of Computer and Information Sciences, CERDAS, Universiti Teknologi PETRONAS (UTP), 32610 Seri Iskandar, Perak Darul Ridzuan, Malaysia

[3] Computing Fundamental Department, FPT University, Hanoi, Vietnam

[4] Faculty of Information and Communication Technology, University Tunku Abdul Rahman, Kampar, Malaysia

[5] Informatics Department, Universitas Islam Indonesia, Yogyakarta, Indonesia

[6] College of Computing and Information Sciences, PAF Karachi Institute of Economics and Technology, Karachi, Pakistan