Lecture Notes on Data Engineering and Communications Technologies 74

Rajeev Tiwari
Mamta Mittal
Lalit Mohan Goyal *Editors*

Energy Conservation Solutions for Fog-Edge Computing Paradigms

Springer

Lecture Notes on Data Engineering and Communications Technologies

Volume 74

Series Editor

Fatos Xhafa, Technical University of Catalonia, Barcelona, Spain

The aim of the book series is to present cutting edge engineering approaches to data technologies and communications. It will publish latest advances on the engineering task of building and deploying distributed, scalable and reliable data infrastructures and communication systems.

The series will have a prominent applied focus on data technologies and communications with aim to promote the bridging from fundamental research on data science and networking to data engineering and communications that lead to industry products, business knowledge and standardisation.

Indexed by SCOPUS, INSPEC, EI Compendex.

All books published in the series are submitted for consideration in Web of Science.

More information about this series at http://www.springer.com/series/15362

Rajeev Tiwari · Mamta Mittal · Lalit Mohan Goyal
Editors

Energy Conservation Solutions for Fog-Edge Computing Paradigms

Springer

Editors
Rajeev Tiwari
Department of Virtualization
University of Petroleum and Energy Studies
Dehradun
Uttarakhand, India

Mamta Mittal
Department of Computer Science
and Engineering
G. B. Pant Government Engineering
College
New Delhi, India

Lalit Mohan Goyal
Department of Computer Engineering
J. C. Bose University of Science
and Technology
Faridabad, Haryana, India

ISSN 2367-4512 ISSN 2367-4520 (electronic)
Lecture Notes on Data Engineering and Communications Technologies
ISBN 978-981-16-3450-5 ISBN 978-981-16-3448-2 (eBook)
https://doi.org/10.1007/978-981-16-3448-2

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

This book focuses on energy efficiency concerns in fog–edge computing and the requirements related to Industry 4.0 and next-generation networks like 5G and 6G. This is the first of its type to guide the research community about practical approaches, methodological, and moral questions in any nations' journeys to conserve energy in fog–edge computing environments. It differs from other published books as it includes a detailed approach required to conserve energy and comparative case studies with respect to various performance evaluation metrics, such as energy conservation, resource allocation strategies, task allocation strategies, VM migration, and load sharing strategies with state-of-the-art approaches, with fog and edge networks. This book has been organized into 14 chapters.

Chapter "Energy-Aware Resource Scheduling in FoG Environment for IoT-Based Applications" presents handling the latency-sensitive applications efficiently, and FoG architecture is widely used with IoT devices to efficiently retrieve or forward the data. For the comprehensive utilization of the resources in the FoG systems based smart cities, various energy-aware resource allocation schemes have been discussed in this chapter. The schemes suggest different mechanisms to access the required contents with minimal energy consumptions for the applications that are used in smart cities and other applications.

Chapter "DoSP: A Deadline-Aware Dynamic Service Placement Algorithm for Workflow-Oriented IoT Applications in Fog-Cloud Computing Environments" offers IoT applications a comprehensive framework that supports QoS-aware service placement in a fog computing environment. This chapter proposes a novel multitier fog computing architecture called Deadline-oriented Service Placement (DoSP) that provides the services in both fog and cloud nodes. This research work proposed a methodology to utilize low-cost fog resources while ensuring that the response time satisfies a given time constraint. It uses the genetic algorithm (GA) to dynamically determine the service placement in the fog environment.

Chapter "Improvement of Task Offloading for Latency Sensitive Tasks in Fog Environment" proposes a QoS-aware task offloading strategy using a novel nature-inspired optimization algorithm, known as the smart flower optimization algorithm (SFOA). The proposed strategy takes into account the QoS parameters such as the task deadlines and budget constraints in selection of appropriate fog nodes where computation tasks can be offloaded. The proposed strategy has been simulated, and the results have verified the efficacy of the strategy.

In Chapter "A Sustainable Energy Efficient IoT-Based Solution for Real-Time Traffic Assistance Using Fog Computing" a Hierarchical Peer- Connected Fog Architecture (HPCFA) have been proposed by authors to lower the latency time and computational overhead. In HPCFA, the fog nodes are organized in a hierarchy. The data from IoT devices equipped on the roads will capture the position of the vehicle, this information is then transmitted to the nearest fog node. This fog node will further transmit the information through HPCFA to the user. Using HPCFA, the total energy consumption is reduced to some extent. The proposed architecture is flexible, as it works both with fog nodes or without fog nodes and directly with the cloud.

In Chapter "Analysis on Application of Fog Computing in Industry 4.0 and Smart Cities," authors have analyzed how fog computing can be effectively utilized to improve Industry 4.0 and smart city applications. A fog computing-enabled architecture helps to reduce some of these challenges by working as a low-complexity computational layer between cloud and Internet of Things (IoT) layers.

Chapter "Fog-Computing: A Novel Approach for Cloud-Based Devices Using Perceptual Cloning Manifestation-PerColNif Taxonomy by Energy Optimization" presents a systematic architecture for timeline-based emergency services to bridge the two platforms, namely cloud server and the edge devices, by supporting computational task through fog cloning over the multiple cloud servers, and thereby ensuring optimal utilization of energy with improved residual sources retained. A novel taxonomy is proposed as a part of minimizing energy consumed for various applications for improvement of QoS.

In Chapter "Performance Evaluation and Energy Efficient VM Placement for Fog-Assisted IoT Environment," authors have presented a multi-server queueing system having reneging with retention policy modeled to measure several performance measures of the fog system. The profit and revenue of the system are analyzed. Further, an efficient greedy-based VM placement scheme GVMP is proposed to optimize the energy consumption of the fog centers. The efficiency of the algorithm GVMP is compared with the state-of-the-art algorithms such as FFD, BFD, RR, and MBFD.

Chapter "Load Balancing in Fog Computing Using QoS" presents a framework (OLBA) for load balancing in fog computing environments to balance the load between fog devices and improves QoS parameters, i.e., turnaround time resource utilization, response time, and delay parameter. This approach is based on particle swarm optimization (PSO) technique to find the local best server and then compare all the local best server selections to find the ultimate global best server selection. Moreover, analysis and comparison with the traditional load balancing techniques is presented.

Chapter "Fog Computing in Industry 4.0: Applications and Chal- lenges—A Research Roadmap" presents an overview of fog computing along with the architectural framework of Industry 4.0. It discusses the various applications of fog computing in industry 4.0. Different problems faced in the implementation of fog computing in Industry 4.0 are discussed. Then, various research challenges are discussed for the efficient deployment of fog in Industry 4.0.

Chapter "Fog Computing Based Architecture for Smart City Projects and Applications" presents a comprehensive literature survey on the deployment architectures of fog computing in smart city applications such as smart waste management and smart parking. An emphasis is laid more on the integration of Industry 4.0's core concepts and fog computing while also taking into consideration the deployment aspects. With the proposed architectures and mentioned approaches, improvements can be seen in terms of resource utilization, processing overhead, and latency. Then, research challenges and future directions are also offered.

In Chapter "Integration of Fog Computing and IoT-Based Energy Harvesting (EHIoT) Model for Wireless Sensor Network," authors have proposed a design model for the integration of a wireless sensor network for the hospital environment by enabling IoT-based energy harvesting (EHIoT) methods. This approach combines collecting healthcare-related data from the hospital environment and monitoring them through the IoT-/fog-based system while applying EHIoT methodologies for harvest energy and management in the proposed WSN.

Chapter "Design and Development of Efficient Secure Routing Mechanism for Wireless Sensor Network" introduces a novel optimization policy to balance the trade-off between energy and security aspects. The experimental simulation of the proposed analytical modeling is carried out in a numerical computing platform with respect to mathematical computation principle. The performance validation of the models, FEESR, DSP-IR, and ORSA, is done with respect to different significant parameters where the comparison has been done with the most significant energy-efficient and secure routing baselines. The simulation outcome outperforms the conventional baselines from both energy and security viewpoints.

Chapter "Futuristic Communication Systems Using Mobile Edge Computing" includes the fundamental characteristics, challenges, and market factors. It shows integration of MEC with the upcoming 5G and beyond technologies including NOMA, WPT EH, UAV, IoT, and heterogeneous CRAN and ML. It addresses the role of MEC in the 5G network architecture. It describes the potential of MECs for future courses.

Chapter "Methodology to Ensure the Continuity of the Information Systems Service, Based on the Monitoring of Electrical Energy, Using IoT Technology" offers methodology that presents a technique based on IoT techniques through which the consumption of the package is monitored of batteries found in the data centers of hospital establishments. This is to analyze the level of charge and the state of the batteries at all times It offers a solution to commercial electrical power, then battery bank get power on, thus tracking on the consumption of the battery Charging solution is offered.

Dehradun, India — Rajeev Tiwari
New Delhi, India — Mamta Mittal
Faridabad, India — Lalit Mohan Goyal

Contents

Editors and Contributors

About the Editors

Dr. Rajeev Tiwari is working as the sr. associate professor in SCS in UPES, Dehradun (India). He is a senior IEEE member. He has done Ph.D. in CSE from Thapar University, Patiala (Punjab). He has more than 16 years of research and teaching experience. His broad areas of research are cloud computing, MANET, VANET, QoS in wireless networks, cache invalidation techniques, Internet of Things (IoT), big data analytics, and machine learning. He is an expert in the simulation and design of scenarios on NS2, SUMO, and MOVE. He is an active member of ACM, IEI, IACSIT, and IAENG. He has 50+ international publications in SCI and Scopus indexed journals and conferences. His publications are majorly indexed in SCI, Scopus, and DBLP. He has published national and international patents also. He has enriched the experience of organizing international conferences successfully. He has major roles as organizing chair, TPC chair, co-chair secretory, etc., in NGCT 2016, NGCT 2017, NGCT 2018, NGCT 2019, ICACCA 2019, IC4S 2019, etc., with Springer and IEEE. He has chaired many sessions in international conferences. He is a lead reviewer in SCI indexed journals of Springer, IEEE, and ScienceDirect and international conferences. He has edited books and published book chapters also with international publishers. He has contributions in funded projects in the field of IoT, cloud and wireless network, FoG/edge computing systems and their QoS.

Mamta Mittal is working in the Computer Science and Engineering Department at G. B. Pant Government Engineering College, Okhla, New Delhi (under Government of NCT Delhi). She earned Ph.D. in Computer Science and Engineering from Thapar University, Patiala; M.Tech. (Honors) in Computer Science and Engineering from YMCA, Faridabad, and B.Tech. in Computer Science and Engineering from Kurukshetra University, Kurukshetra, in 2001. She has been teaching from the last 18 years with emphasis on data mining, machine learning, soft computing and data structure. Dr. Mittal is a lifetime member of CSI and published and communicated more than 90 research papers in SCI, SCIE, and Scopus indexed journals. She holds five patents and two copyrights granted and three patents published in the areas of artificial intelligence, IoT, and deep learning. Dr. Mittal has edited/authored many books with reputed publishers like Springer, IOS Press, Elsevier, and CRC Press and working on DST approved Project “Development of IoT based hybrid navigation module for mid-sized autonomous vehicles” with research grant of 25 Lakhs. Currently, she is guiding Ph.D. scholars in machine learning, computer vision and deep learning areas. Dr. Mittal is the editorial board member with InderScience, Bentham Science, Springer, and Elsevier, handled special issues and has chaired a number of conferences. She is the book series editor of Health Informatics and Health care, using AI and smart computing and another series edge AI in future computing with CRC Press, Taylor & Francis, USA.

Dr. Lalit Mohan Goyal has completed Ph.D. from Jamia Millia Islamia, New Delhi, in Computer Engineering, M.Tech. (Honors) in Information Technology from Guru Gobind Singh Indraprastha University, New Delhi, and B.Tech. (Honors) in Computer Engineering from Kurukshetra University, Kurukshetra. He has 17 years of teaching experience in the areas of theory of computation, parallel and random algorithms, distributed data mining, and cloud computing. He has completed a project sponsored by the Indian Council of Medical Research, Delhi. He has published many research papers in SCI indexed and Scopus indexed journals and conferences. Two patents filed by him are also now published online. He is a reviewer of many reputed journals and conferences. He has given many invited talks in FDP and conferences. Presently, he is working at J. C. Bose University of Science and Technology, YMCA, Faridabad, in the Department of Computer Engineering.

Contributors

Mahmoud A. L. Ahmad School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar, India

Wilver Auccahuasi Universidad Privada del Norte, Lima, Perú

Rabindra Kumar Barik School of Computer Applications, KIIT Deemed to be University, Bhubaneswar, India

Nicanor Benites Univeraidad Nacional Mayor de San Marcos, Lima, Perú

Rajkumar Buyya Cloud Computing and Distributed Systems (CLOUDS), Laboratory, School of Computing and Information Systems, The University of Melbourne, Melbourne, Australia

Meetali Chauhan Department of Computer Science and Engineering, Gulzar Group of Institutes, Khanna-141401, Punjab, India

Priyanka Chawla School of Computer Science, Lovely Professional University, Phagwara, Punjab, India

L. M. Darshan Department of CS & E, Adichunchanagiri Institute of Technology, Chikkamagluru, Karnataka, India;
Department of CS & E, Government Engineering College, Kushalnagar, Karnataka, India

D. Deepa Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India

Monica Diaz Universidad Norbert Wiener, Lima, Perú

Edwin Felix Universidad Nacional de Moquegua, Moquegua, Perú

Edward Flores Universidad Nacional Federico Villarreal, Lima, Perú

Shelly Garg School of Computer Science, University of Petroleum and Energy Studies (UPES), Dehradun, Uttarakhand, India

Sukhpal Singh Gill School of Electronic Engineering and Computer Science, Queen Mary University of London, Mile End, UK

Shilpi Harnal Institute of Engineering and Technology, Chitkara University, Punjab, India

H. M. K. K. M. B. Herath Faculty of Engineering Technology, The Open University of Sri Lanka, Nugegoda, Sri Lanka

D. Jude Hemantha Department of Electronics and Communication Engineering, Karunya University, Coimbatore, Tamil Nadu, India

Aman Kataria CSIR-CSIO, Chandigarh-160030, India

Maninder Jeet Kaur Department of Applied Science and Engineering Technology, Seneca College, Toronto, Canada

Parmeet Kaur Department of Computer Science & Enginnering and Information Technology, Jaypee Institute of Information Technology, Noida, India

B. Keerthi Samhitha Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India

Rupa Kesavan DCSE, Sri Venkateswara College of Engineering, Pennalur, Sriperumbudur, India

Sumit Kumar School of Computer Science, University of Petroleum and Energy Studies (UPES), Dehradun, Uttarakhand, India

Sunny Kumar Department of Electronics and Communication Engineering, Bhagwan Parshuram Institute of Technology, GGSIPU, Delhi, India

Vijayaraja Loganathan DEEE, Sri Sairam Institute of Technology, Chennai, India

Denny Lovera Universidad Nacional Federico Villarreal, Lima, Perú

B. G. D. A. Madhusanka Faculty of Engineering Technology, The Open University of Sri Lanka, Nugegoda, Sri Lanka

Piyush Maheshwari Faculty of Engineering and IT, The British University, Dubai, United Arab Emirates

Suja Cherukullapurath Mana Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India

Naishadh Mehta Institute of Technology, Nirma University, Ahmadabad, India

Shikha Mehta Department of Computer Science & Enginnering and Information Technology, Jaypee Institute of Information Technology, Noida, India

Ravi Dutt Mishra Seth Jai Parkash Mukand Lal Institute of Engineering and Technology, Radaur, Haryana, India

Mamta Mittal Department of Computer Science and Engineering, G. B. Pant Government Engineering College, South East Delhi, New Delhi, India

Arif Mushtaq Faculty of Management Information System, City University College of Ajman, Ajman, United Arab Emirates

Orlando Pacheco Universidad Nacional Federico Villarreal, Lima, Perú

Sudhansu Shekhar Patra School of Computer Applications, KIIT Deemed to be University, Bhubaneswar, India

J. K. Periasamy DCSE, Sri Sairam Engineering College, Chennai, India

R. D. D. Prematilake Faculty of Science and Technology, University of Plymouth, Plymouth, UK

Sita Rani Department of Computer Science and Engineering, Gulzar Group of Institutes, Khanna-141401, Punjab, India

Sadia Riaz Faculty of Information Technology and Research Methods, S. P. Jain School of Global Management, Dubai, United Arab Emirates

Luis Romero Universidad Nacional Federico Villarreal, Lima, Perú

H. R. Roopashree Department of CS & E, Adichunchanagiri Institute of Technology, Chikkamagluru, Karnataka, India

Mario Ruiz Universidad Nacional de Moquegua, Moquegua, Perú

Anand Ruparelia Institute of Technology, Nirma University, Ahmadabad, India

Fernando Sernaque Universidad César Vallejo, Lima, Perú

Nidhi Seth Seth Jai Parkash Mukand Lal Institute of Engineering and Technology, Radaur, Haryana, India

T. Shankar DECE, Government College of Engineering—Srirangam, Trichy, India

Gaurav Sharma Seth Jai Parkash Mukand Lal Institute of Engineering and Technology, Radaur, Haryana, India

Meeniga Sriraghavendra School of Computer Science, Lovely Professional University, Phagwara, Punjab, India

C. K. Subbaraya Adichunchanagiri University, Javarana Hally, Karnataka, India

Sumit Department of Electronics and Communication Engineering, Bhagwan Parshuram Institute of Technology, GGSIPU, Delhi, India

Bhawna Suri Department of Computer Science and Engineering, Bhagwan Parshuram Institute of Technology, GGSIPU, Delhi, India

Shweta Taneja Department of Computer Science and Engineering, Bhagwan Parshuram Institute of Technology, GGSIPU, Delhi, India

N. L. Taranath Department of CS & E, Adichunchanagiri Institute of Technology, Chikkamagluru, Karnataka, India

Rajeev Tiwari School of Computer Science, University of Petroleum and Energy Studies (UPES), Dehradun, Uttarakhand, India

Kitty Urbano Universidad Privada del Norte, Lima, Perú

Jai Prakash Verma Institute of Technology, Nirma University, Ahmadabad, India

R. Vignesh Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India

Huaming Wu Center for Applied Mathematics, Tianjin University, Tianjin, PR China

A. C. Yogeesh Department of CS & E, GSSSITEW, Mysuru, Karnataka, India

Energy-Aware Resource Scheduling in FoG Environment for IoT-Based Applications

Rajeev Tiwari, Mamta Mittal, Shelly Garg, and Sumit Kumar

Abstract Internet of Things (IoT) has been developed as a heterogeneous environment that contains network devices with limited resources. The application of IoT principles in the smart city domain creates new opportunities and requires diligent implementation mechanisms for optimal resource utilization. With time, the IoT applications tend to generate and forward a huge amount of data in the smart cities and require a real-time response from the servers. Due to this, the traditional cloud computing architecture is unable to handle the latency-sensitive applications efficiently, and hence, the FoG architecture has been widely implemented with IoT devices to efficiently retrieve or forward the data. For the comprehensive utilization of the resources in the FoG systems-based smart cities, various energy-aware resource allocation schemes have been discussed in this chapter. The schemes suggest different mechanisms to access the required contents with minimal energy consumptions for the applications that are used in smart cities.

1 Introduction

In recent times, usage of Internet-enabled things has changed the way of thinking and working of people. It has brought the concept of being connected anything-anywhere with anyone at any time. This evolution has attracted many researchers in this domain. IoT [1–3] contains sensors, actuators and the devices having capabilities of connecting via the Internet called as smart devices. These devices have sensing

R. Tiwari · S. Garg (✉) · S. Kumar
School of Computer Science, University of Petroleum and Energy Studies (UPES), Dehradun, Uttarakhand, India
e-mail: shelly@ddn.upes.ac.in

R. Tiwari
e-mail: rajeev.tiwari@ddn.upes.ac.in

M. Mittal
Department of Computer Science and Engineering, G. B. Pant Government Engineering College, Okhla, South East Delhi, New Delhi, India

R. Tiwari et al. (eds.), *Energy Conservation Solutions for Fog-Edge Computing Paradigms*, Lecture Notes on Data Engineering and Communications Technologies 74,
https://doi.org/10.1007/978-981-16-3448-2_1

capabilities and perform the task of data collection. This data is processed on the servers for accomplishing different tasks. These features are suitable for the smart cities that also require different types of sensors to collect the data across the city and utilize this data to manage the assets, resources and services in an efficient manner. The data can be collected from citizens, industries, institutes, roads, etc. and analyzed to observe and control the transport, institutional activities, gas plants, power plants, waste management and other utility services across the city. However, as the number of IoT devices is increasing exponentially, the amount of data is also rising that results in growing energy needs. For various researchers, energy consumption prediction and optimization has come out to be a major area of concern.

For the applications of smart city, G-IoT (Green-IoT) [4–6] are introduced to overcome the energy shortage. The traditional methods to overcome energy shortage are production of more energy, which is a costly solution making it of least interest and the other is to use the renewable energy resources that also have several limitations. There is one more promising solution that minimizes the energy consumption using the pre-emptive measures which are discussed in this chapter. With increase in the data amount, the requirement of an efficient mechanism to store, process and data retrieval has emerged because the connected objects use different energy utilization strategies.

In this direction, cloud computing [7–10] is the initial platform for IoT business applications that allow individuals and organizations to utilize the resources at reasonable cost and enhance its usage on a broad perspective. In the cloud computing architecture, the edge devices forward their data to centralized cloud servers for processing tasks. After the processing, the cloud servers send the requested data in the reverse direction toward edge devices. However, due to the increase in the number of IoT devices and their processing needs, the cloud computing architecture is unable to provide real-time response to latency-sensitive applications in smart cities. To ease the load of cloud servers and network traffic, various caching schemes have also been suggested for the IP-based [11–14] and emerging Internet architectures such as Content-Centric Networks (CCN) [15–17]. But, the heterogeneity and availability of resources have generated the need of efficient utilization of resources for sustainable smart cities.

In 2011, CISCO brought a new concept called FoG computing to resolve the limitations of cloud architecture. The FoG architecture [18–23] is popularly defined as the distributed network to provide speedy services amid the cloud and geographically distributed end user devices to bid computation offloading prospects such as latency, bandwidth, and energy consumption. The efficient utilization and management of resources can be achieved using technologies such as virtualization and containers. Cloud computing is majorly grounded on using high capacity data centers whereas FoG computing uses FoG nodes which is a set of heterogeneous and distributed resources with moderate storage and processing capabilities. FoG computing has proven to be an efficient architecture to handle the requirements of latency-sensitive applications with minimal bandwidth and energy requirements. In the smart cities, a large number of requests for resources has been generated and FoG computing

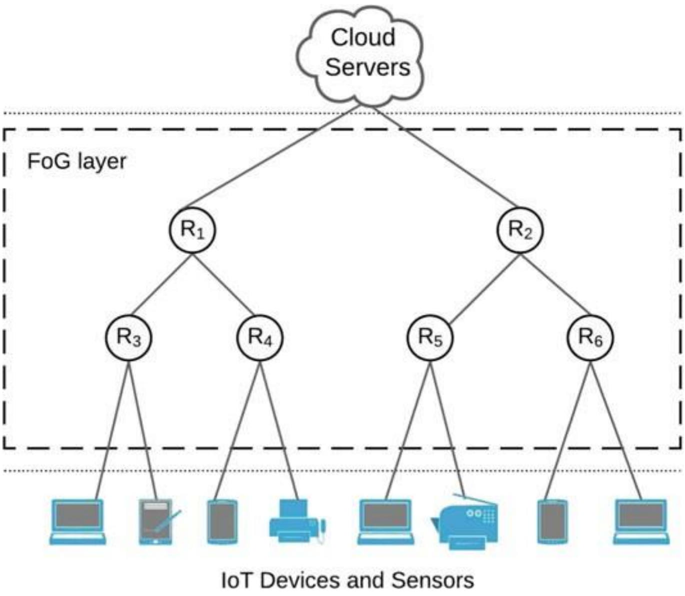


Fig. 1 FoG architecture with cloud server, fog nodes and IoT devices

has come out to be a promising system to provide real-time processing for the smart cities applications.

The FoG architecture contains three layers as shown in Fig. 1 consisting of cloud layer on the top, FoG layer in the middle and all the edge devices (IoT devices) in the end [24]. The architecture provides processing of application's data near the edge devices and significantly reduces the traffic and energy requirements within the network. The data generation occurs in the edge devices, where data sensing happens using actuators, sensors and similar devices. Then, the requests are further forwarded to the FoG layer based on that data. If the requested data is available within the FoG node, then response for the request is generated within the FoG layer. Otherwise, the request is satisfied by the cloud servers. Both cloud and FoG systems are based on distributed approaches but cloud computing is based on the centralized approach whereas, the FoG follows a decentralized approach. In a FoG environment, any intermediary node having computational and storage capabilities can act as a FoG node. However, limitation and heterogeneity of resources have made a challenging task of resource allocation and provisioning in the FoG layer.

1.1 Why FoG Computing?

FoG computing comes out to be an efficient way to enhance the Quality-of-Service (QoS) for the end IoT devices such as sensors, actuators, smart cameras, smartphones,

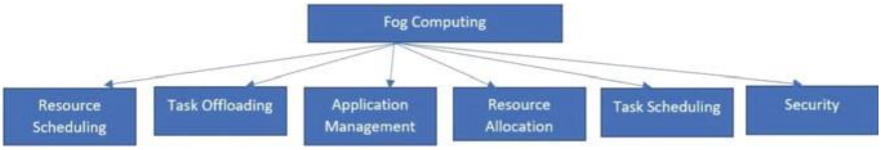


Fig. 2 Taxonomy of FoG computing

and other devices in IoT. These devices consist of very limited capabilities which is not sufficient for the current Internet applications that require more powerful devices. One potential solution to this problem is to use the cloud servers in such applications. However, it increases the latency because of the far-reaching network path among the cloud data centers and IoT devices. Moreover, this might increase the bandwidth usage that results in increased congestion probability within the network. The solution to this problem is FoG computing as it provides the solution for the latency and network traffic issues. Figure 2 demonstrates the FoG computing taxonomy [25] based on the following aspects:

Cloud computing limitations for G-IoT (distributed deployment)

Resource scheduling: Depending on the requests from the applications, the resource scheduling plans the resources where FoG services would be executed.

Task offloading: For optimal resource utilization, traditional scheduling schemes may not be a suitable approach. In this case, some of the requests can be offloaded to other FoG nodes to achieve the desired QoS for the end users.

Application management: Application management defines the placement and management of applications in the FoG network as optimal application placement improves the utilization of network resources.

Resource allocation: Resource allocation refers to the assignment of resources to the applications in the FoG architecture.

Task scheduling: It determines the FoG-based application that would utilize the FoG layer for its execution. The traditional task scheduling schemes are FIFO (First-In-First-Out), SJF (Shortest-Job-First), Roundrobin, etc.

Security: It handles the security-related issues that can arise in FoG-based computing architecture.

1.2 *FoG Application Areas in Smart City*

FoG computing is widely used in the domains where large amounts of data need to be handled on a daily basis that includes wind energy, transportation, smart buildings, smart cities, security purposes, etc. In day-to-day life, the cities are facing a large set of problems such as traffic, security, and efficient utilization of resources [21]. One proposed solution to such problems is to install the IoT-based FoG network in the computing architecture, which consists of FoG nodes. These nodes offer local storage and processing facilities in the smart cities and improve QoS for the citizens who face the problems of shortage of broadband and connectivity because the number of customers is increasing on a daily basis. Toward this, some of the applications of Fog architecture in smart cities are as follows:

Smart Homes Appliances: The IoT-based home appliances are connected to each other and with the Internet to improve the ease of living of citizens in smart cities. These large numbers of devices across the city, generate requests for different types of data and lead to congestion in the entire network. The FoG system is used with these devices to ease the data processing and retrieve the requested data without accessing the servers and thus, reduces propagation delay and energy requirements.

Security: To enhance the security, video cameras are used across the city and used in observing the activities. This huge amount of multimedia data requires real-time monitoring and processing with low latency. Using the FoG architecture, video processing is divided among the FoG nodes that are placed between cloud data centers and the end device, i.e., cameras.

Transport: Today the vehicle manufacturing companies are releasing self-driving vehicles that are connected to the Internet and nearby vehicles. Therefore, the IoT devices embedded in those vehicles produce large amounts of data such as traffic information, routing information, climatic conditions, and vehicle condition. This data requires real-time processing which is performed by the FoG computing systems such as FoG nodes continuously update the information about the vehicle condition to its manufacturers for maintenance purposes.

Real-Time Data Transfer and its Processing: Data transfer from one place to another is achieved using FoG deployments as these nodes are used to transfer the data within the smart city. For example, the transfer of a crime information from one police station to another police station with minimal delay. Many smart city applications such as data streaming, customer service systems require real-time data processing which can be performed optimally using Fog systems [26, 27].

Waste Management: The IoT-based containers transmit the garbage level information to the connected garbage collection vehicles using FoG nodes. The vehicles then optimize the routes based on the garbage level information.

Air Pollution: In the smart cities, IoT devices are placed at various locations across the city to measure the carbon level, emission from factories, vehicles and toxic gases generated in the farms. This large data requires decentralized processing which is achieved using FoG systems.

1.3 Motivation

Many of the smart IoT applications such as smart home, smart city, and smart healthcare management need to handle dynamic resource management and efficient utilization for a sustainable environment. To accomplish this, FoG computing is a promising paradigm as FoG layer serves between cloud layer and edge layer provides a favorable computing environment which expands the concept of networking, computing and gives storage services for data centers and end users. All the sensitive applications in terms of location, bandwidth, time consumption are being handled by FoG computing for IoE (Internet of Everything). Consider an example of a smart transportation system in a smart city where an intelligent vehicle is traveling on a road to reach its destination point. Any event such as congestion, accident or any kind of environmental issues can happen on the way. The system should be capable enough of detecting and handling those issues which can further lead to change in user requirements. In case of any kind of failure can lead to delay in reaching the destination which can further result in loss and damages. As per above scenario discussed, due to dynamic nature of user, management of resources dynamically with energy conservation schemes is the need of the hour.

1.4 Contributions

This chapter focuses on resource allocation and provisioning schemes for the green-IoT in the smart cities [28, 29]. To the best of our knowledge, various researchers have surveyed in green-IoT fields which are taken into account and are also discussed thoroughly in the related work section of this chapter. The key contributions of this chapter are as follows: The detailed discussion on the need and application of FoG computing in the smart city. Then, the design of the FoG computing taxonomy has been explained in the IoT environment for smart city. Finally, various energy-aware algorithms are elaborated for the sustainable smart city and their future scope has been described.

1.5 Organization of Work

The rest of the chapter is organized as follows,

Section 2 Discusses the need of FoG computing with all the Quality-of-Service (QoS) parameters and contribution of those parameters in the smart cities.

Section 3 Presents a tabular description of energy-aware survey focusing on green-IoT with a thorough discussion of scope and observation for sustainable smart cities

Section 4 Explains the resource scheduling algorithms with energy conservation concept.

Section 5, Presents the intersection of parameters for various resource scheduling algorithms like for every algorithm addressing of parameters are done such as future energy prediction (FEP), energy optimization (EO), prizing schemes such as Real Time Pricing (RTP), scheduling and cost reduction parameters such as reduce power consumption (RPC) and reduce cost (RC) in case of Smart Homes scenario in Smart cities.

Section 6 discusses the future scope of the work.

2 Need of FoG Computing

In a network, all end user devices are placed on the bottom level which includes smartphones, sensors, actuators, etc. All the data retrieved via end user devices brought for further processing in the FoG layer for data processing and storage. A centralized approach is provided by the cloud but this has increased the challenge of data processing and management, which rises to the term "FoG computing" In this architecture, operations are carried out majorly between the cloud layer and end devices [30]. All edge devices work hand in hand with the FoG nodes and the servers. Table 1 discusses the difference between computing techniques, i.e., cloud, FoG and edge.

2.1 QoS Parameters

Various angles in FoG computing, such as diverseness, mobility and distribution, QoS is a big issue in FoG computing. FoG computing is part of a large variety of QoS factors such as throughput, deadline, response utilization, cost, execution time, response time, energy consumption, security, availability and scalability. The parameters that are also a part of objectives have been discussed and their definitions are explained as follows:

Table 1 Dissimilarity between cloud, FoG and edge computing

Parameter	Cloud computing	FoG computing	Edge computing
Placement in model	Topmost layer in the architecture	Middle layer of the architecture	Bottom layer of architecture
Contains	Cloud data centers	FoG nodes/servers	Sensors, controllers, smart devices
Usage	Big Data Processing with data warehousing	Data analysis on local network with a control response	Large volume real-time data processing via applications

Latency (Time): Time delay happened between the change of state of any task.

Cost: Cost of execution, cost of network utilization.

Energy: Energy consumed for execution, transfer purpose.

Bandwidth: Data rate calculation between the edge-FoG or FoG-cloud nodes.

Profit: Profit gained because of FoG or cloud nodes or both.

Minimizing Traffic: Total traffic minimization in the network.

Security: Measures provided by FoG nodes to enhance the security.

Resource Utilization: Maximum use of available resources.

Though put: Maximum solution required that can have a cost in the system.

Scalability: The capacity of an information system is such that, by increasing the number of service requests or the use of resources, the performance of the system is maintained.

3 Energy-Aware Survey in Green-IoT

Green-IoT [6, 8, 31] is a term which talks majorly about sustainable development in IoT in terms of energy which has become the need of the hour as a number of devices are increasing day by day and leading to huge amount of generation of data and increasing the energy requirements. Several researchers have done related work with major focus on IoT and sustainable smart city. Energy conservation has become one of the most attractive areas of research for researchers to make the world a better place with the efficient use of technology. Table 2 discusses year-wise energy-related work specifying about the areas mentioned, discussed and observations gathered from the existing work. Many of the work discusses IoT in a wide spectrum but work discussed in [31–35] solely focused upon efficiency of energy in IoT aiming at technologies in networking, energy communication and energy harvesting, applications in green industrial applications, how power is deployed and managed in G-IoT. Surveys conducted in 2019 talks about Erlang, a unit used to measure traffic density in telecommunication systems where one call is measured equivalent to one Erlang [33]. Green energy conservation and wireless charging are also a matter of interest in the energy aware field [34]. For sustainable development in the field of smart cities many of research work is being conducted in Cloud, FoG and edge discussion various QoS parameters such as latency, time delay, cost, energy conservation etc. [35].

Table 2 Energy-aware survey in green-IoT

References	Year	Areas addressed	Observations
[4]	2015	Access networks and cellular network consideration in IoT	Perks of cellular network in IoT and potential energy efficient approaches
[8]	2015	Different issues and technology discussion for green-IoT for a smarter world	IoT central technologies, WSN, RFID, cloud computing, data centers, and machine to machine communication are introduced
[9]	2018	Solutions for green-data centers, communication network, machine to machine, wireless sensor network, cloud computing, Internet are discussed	Different schemes are discussed leading toward green-ICT
[32]	2014	Challenges, research directions and industrial IoT applications	IoT applications in industries are thoroughly covered
[31]	2015	IoT central energy conservation, efficient energy utilization for pollution issues	Introduction to energy central recent schemes for smart cities
[33]	2019	Erlang performance model with enhancement in performance is discussed	For all the incoming requests, probability is to get the delays in network
[34]	2019	Green energy conservation and balancing and wireless charging	No cost green energy to power IoT devices and enabling of wireless charging of devices
[35]	2020	Energy saving techniques in cloud, FoG, edge	Various QoS parameters are discussed

Review this later

Review this later

Review this later

4 Related Work

FoG Computing [22–25, 36] has come up as a promising platform to provide solutions to resource scheduling issues due to diverse nature of resources as well as limited availability. It is still getting explored to get the maximum benefits out of it. However, some of the work has been done in the past few years. Kenyon and Cheliotis et al. [37] has proposed a solution to dynamic resource requirements for users for grid resources considering resource characteristics and delivery, that framework contains a group of users and another one as external providers. However, no experimental scenarios are discussed for validation purposes. Schlagkamp et al. [38] diverseness and similarities in job submission behavior of users in HPC (High Performance Computing) and HTC (High Throughput Computing).

Drawback...

4.1 Resource Allocation and Provisioning in Cloud Computing

Many resource allocation algorithms with energy optimization techniques are proposed for multi-cloud environments. In [39], researchers have proposed a novel task allocation scheme using the make span scheme for energy optimization. Kendrick et al. [31] with the usage of brute force technique, researcher has proposed a multi-agent and multi-cloud-based technique where if services do not fit in one cloud data center, then they approach other clouds using memory-based approach to fulfill the service requirements. In [41], researchers have introduced a network slicing technique for service allocation in 5G network based on adjustment of parameters priority basis but dynamic user requirements are not considered and hence giving a scope to do further research. Work has been done in this field but as cloud computing is more of a centralized approach, now the inclination has increased more toward the decentralized approach which can be achieved by FoG computing [40].

What is memory based approach in cloud computing?

4.2 Resource Allocation and Provisioning in Fog Computing

Placement of applications for various scenarios has been studied by various authors. In the digital world, the government has an agenda to make cities smart which is based on the concept of "connected anywhere-anyplace-anytime" faces some challenges such as network utilization in terms of energy, time, execution cost, and bandwidth. Toor et al. [15] address the energy-aware scheme where he discusses that using the renewable and nonrenewable resources of energy on FoG data centers can result in better utilization of resources but he still faces the challenge of efficient utilization of energy in conservative mode and as per him a deep learning solution can be proposed to solve this problem. Jalali et al. [16] propose a technique based on energy consumed by Cloud and IoT devices using temperature sensors, PIR sensors and camera were connected to Raspberry Pi which is further connected to Watson IoT platform via Ethernet, her work has considered FoG computing and microgrid setup taking weather conditions into account and local renewable energy, but still, there is an issue of automatic decision making, proposed solution can be usage of machine learning techniques. Developing smart cities in the main agenda nowadays and with the increase in demand, many of the challenges are faced which are addressed by researchers and increase the scope of work in this field. Wan et al. [17] proposed ELBS algorithm for industry 4.0 has its own advantages but dynamic randomness in actual scheduling, fuzzy delivery data, uncertain processing time and emergency order need to be processed is clearly stating a gap where various schemes still need to be improved and generating a wide range of research option to for-coming researchers to work in this field. Alzeyadi et al. [18] extended Wan et al. [17] work and proposed an improvised ELBS algorithm considering factors such as energy network efficiency, traffic and latency with decrease in network rate by 63%.

Naranjo et al. [22] analyze the domain of smart city where the authors pulled up a strategy named FoG Computer architecture network (FOCAN) where the applications are set to compute, route and communicate in an amalgamated fashion but as the whole FOCAN is interconnected, security breach may occur.

Sun et al. [34] give a resource scheduling procedure for management of heterogeneous resources. This scheme has used the NSGA-2 algorithm on the MATLAB simulation environment. This algorithm is based on an improvement basis on non-dominated sorting genetic algorithms. This scheduling is based on multiobjective optimization in FoG environments resulting in improvement in stability of task execution and reduction in service latency [20]. Table 3 discussed the advantages and technology used by various researchers in the field of energy conservation with simulation environment is discussed in detail.

Table 3 Related work table

[No.]	Author name	Year	Simulator used/setup used	Technique used	Advantages
[15]	Toor et al.	2019	iFoGSim	Renewable and non renewable energy; DVFS (Dynamic Voltage and Frequency Scaling)	Charging of nodes are considered mainly to reduce the limitation of battery capacity, energy consumption still can be improved, in conservative mode, a deep learning solution need to be proposed
[16]	Jalali et al.	2016	Temperature sensor, PIR sensor and cameras were connected to Raspberry Pi; then Pi is connected to Watson IoT platform via ethernet	Energy consumed by cloud and IoT setup	Considered FoG computing and micro grid setup taking weather conditions into account and local renewable energy
[17]	Wan et al.	2018	Raspberry pieboard, UDOO board, ESP8266	ELBS algorithm (Energy aware Load Balancing and Scheduling)	Proposed method has important role in equipment fault diagnosis and task transfer
[18]	Alzeyadi et al.	2019	iFoGSim	Extended ELBS factors such as energy network efficiency traffic and latency is considered	Network utilization rate is decreased by 63%

5 Energy Resource Scheduling Algorithms and Intersection of Parameters in Smart City (Smart Homes)

This section discusses one of the major research problem and frequently asked question that which algorithm, techniques, parameters and pricing schemes are being used for energy harvesting in smart cities and to give answer to this question a case study of smart home in smart city is taken and basis of that case study various energy optimization techniques are studied and a comparative analysis is done in that scenario. Table 4 [10, 17, 18, 22, 34, 35, 37, 38] addresses the Energy resource scheduling algorithms like PSO, various genetic algorithms, MILP, DMES etc. focused on energy saving and cost reduction. Table 5 [33, 42–46] addresses the Dynamic Programming algorithm, EDE, MPC, genetic algorithm focused on energy saving and cost reduction and Table 6 [38, 47–51] and Table 7 [49, 50, 52–55] focuses on optimized home energy management systems, renewable energy resources and various optimization techniques with discussion on parameters such as algorithm or technique used with area of focus. Energy optimization comprising of following parameters has been discussed considering the scenario of smart homes in smart city with two options 'Y' stating as yes and 'N' stating as NO, parameters are discussed as follows:

Future energy prediction (FEP): consumption of energy is evaluated and prediction is done on future consumption

Energy Optimization (EO): act of effective utilization of energy is done

Real-time pricing (RTP): per unit cost consumption is taken into account when price is high hence to lower the energy consumption

Reduce power consumption (RPC): energy consumption is reduced to minimize greenhouse effect in smart city

Reduced cost (RC): cost of energy consumption is reduced aiming to increase the profitability.

6 Future Scope

Many energy-aware and optimization schemes have been proposed but there is still a clear gap of introducing more energy conserved techniques as per demand. In conservation mode, still energy harvesting needs to be done which can be done by proposing a deep learning solution. Automated decision-making systems can be designed for conservation of energy, which can be achieved by applying machine learning techniques. Also, proposed ELBS algorithm for industry 4.0 has its own advantages but dynamic randomness in actual scheduling, fuzzy delivery data, uncertain processing time and emergency order need to be processed is clearly stating a gap where various schemes still need to be improved and generating a wide range of research option to for-coming researchers to work in this field.

Table 4 Energy resource scheduling algorithms and intersection of parameters in smart city (smart home)

Ref. No.	Algorithm	Focus area	FEP	EO	RTP	RPC	RC	N/w utilization	Latency	Security
[10]	Genetic algorithm, earthworm optimization algorithm	Energy saving, reduce cost	N	Y	N	Y	Y	N	N	N
[17]	Energy aware Load Balancing and Scheduling algorithm (ELBS)	Energy saving	Y	Y	N	Y	N	N	N	N
[18]	Extended-ELBS (improved PSO and ELBS) algorithm	Energy saving	Y	Y	N	Y	N	Y	Y	Y
[22]	FOCAN algorithm	Cluster based fog, peer to peer network	N	Y	N	N	N	Y	Y	Y
[34]	NSGA-2 (non-dominated sorting genetic algorithm)	Energy saving and latency	N	Y	N	N	N	N	Y	N
[37]	Particle Swarm Optimization (PSO), Action Dependent Heuristic Dynamic Programming (ADHDP), Back Propagation (BP), neural network	Energy saving and reduction in cost	Y	Y	Y	Y	Y	N	Y	N
[38]	Daily Maximum Energy Scheduling (DMES), mixed integer linear programming	Energy saving, reduce cost	N	Y	N	Y	Y	N	N	N

Energy-Aware Resource Scheduling in FoG Environment … 13

Table 5 Energy resource scheduling algorithms and intersection of parameters in smart city (smart home)

Ref. No.	Algorithm	Focus area	FEP	EO	RTP	RPC	RC	N/w Utilization	Latency	Security
[33]	Dynamic programming algorithm, regression algorithm, recurrent neural network, support vector regression, random forest regression algorithm	Energy saving, reduce cost	Y	Y	Y	Y	Y	Y	Y	N
[42]	Bacterial Foraging Optimization Algorithm (BFOA), Flower Pollination Algorithm (FPA)	Energy saving, reduce cost	N	Y	Y	Y	Y	N	Y	N
[43]	Optimized Home Energy Management System (OHEMS), Genetic Algorithm (GA), Binary Particle Swarm Optimization (BPSO), Wind-Driven Optimization (WDO), Hybrid GAPSO (HGPO), Multiple Knapsack Problem (MKP)	Energy saving, reduce cost	Y	Y	N	Y	Y	Y	N	N
[44]	Genetic Algorithm (GA), Teacher Learning-Based Optimization (TLBO), Linear Programming (LP)	Energy saving, reduce cost	N	Y	N	Y	Y	N	Y	N
[45]	Hammerstein-Wiener model, model predictive control, Temperature Setpoint Assignment (TSA) algorithm	Reduce cost, energy saving	Y	Y	Y	Y	Y	N	N	N
[46]	Genetic algorithm	Energy saving and reduce cost15	N	Y	N	Y	Y	N	Y	N

Table 6 Energy resource scheduling algorithms and intersection of parameters in smart city (smart home)

Ref. No.	Algorithm	Focus area	FEP	EO	RTP	RPC	RC	N/w utilization	Latency	Security
[47]	Earthworm optimization algorithm, bacterial foraging algorithm	Energy saving and reduce cost	N	Y	N	Y	Y	N	N	N
[48]	Genetic algorithm and Artificial Fish Swarm Optimization (APSO)	Energy saving, reduce cost	N	Y	Y	Y	Y	N	N	N
[38]	Mixed Integer Linear Programming (MILP), Daily Maximum Energy Scheduling (DMES)	Energy saving and reduce cost	N	Y	N	Y	Y	N	Y	N
[49]	Sequential quadratic programming, Levenberg-Marquardt, interior-point	Energy saving, reduce cost	N	Y	Y	Y	Y	N	N	N
[50]	Enhanced Differential Evolution (EDE), Teacher Learning-Based Optimization (TLBO)	Energy saving, reduce cost	Y	Y	Y	Y	Y	N	N	N
[51]	Model Predictive Control (MPC)	Energy saving, reduce cost	Y	Y	N	Y	Y	N	N	N

Table 7 Energy resource scheduling algorithms and intersection of parameters in smart city (smart home)

Ref. No.	Algorithm	Focus area	FEP	EO	RTP	RPC	RC	N/w utilization	Latency	Security
[49]	Energy Consumption Scheduling (ECS) device, distributed algorithm	Energy saving, reduce cost	Y	Y	N	Y	Y	N	N	Y
[52]	Dynamic programming, BPSO, hybrid scheme GAPSO, Multiple Knapsack Problem (MKP)	Energy saving, reduce cost	Y	Y	N	Y	Y	N	N	N
[53]	GA, Cuckoo Search Optimization Algorithm (CSOA), Smart Electricity Storage System (ESS)	Energy saving, reduce cost	N	Y	Y	Y	Y	Y	Y	N
[54]	Home Energy Management System (HEMS), Energy Storage System (ESS), Renewable Energy Resources (RES), Earliglow-based optimization method, Jaya algorithm, enhanced differential evolution, strawberry algorithm (SBA)	Energy saving, reduce cost	N	Y	N	Y	Y	N	N	N
[55]	Genetic Harmony Search Algorithm (GHSA), Wind-Driven Optimization (WDO), Harmony Search Algorithm (HSA), Genetic Algorithm (GA)	Energy saving, reduce cost	N	Y	Y	Y	Y	N	Y	N
[50]	GA (Genetic Algorithm) PSO (Particle Swarm Optimization) Bees-Swarm optimization	Energy saving and La-17 tency	Y	Y	N	N	N	N	Y	N

References

1. Mahmoud MM, Rodrigues JJ, Saleem K, Al-Muhtadi J, Kumar N, Korotaev V (2018) Towards energy-aware fog-enabled cloud of things for healthcare. Comput Electr Eng 67:58–69
2. Khan ZA, Butt AA, Alghamdi TA, Fatima A, Akbar M, Ramzan M, Javaid N (2019) Energy management in smart sectors using fog based environment and meta-heuristic algorithms. IEEE Access 7:157254–157267
3. Shahid MH, Hameed AR, Islam SU, Khattak HA, Din IU, Rodrigues JJ (2020) Energy and delay efficient fog computing using caching mechanism. Comput Commun
4. Laya A, Alonso L, Alonso-Zarate J, Dohler M (2015) Green MTC, M2M, internet of things. Green Communications: Principles, Concepts and Practice, pp 217–236
5. Shaikh FK, Zeadally S, Exposito E (2015) Enabling technologies for green internet of things. IEEE Syst J 11(2):983–994
6. Tuysuz MF, Trestian R (2020) From serendipity to sustainable green IoT: technical, industrial and political perspective. Comput Netw 182:107469
7. Butt AA, Khan S, Ashfaq T, Javaid S, Sattar NA, Javaid N (2019) A cloud and fog based architecture for energy management of smart city by using meta-heuristic techniques. In: 2019 15th International wireless communications & mobile computing conference (IWCMC). IEEE, pp 1588–1593
8. Zhu C, Leung VC, Shu L, Ngai EC-H (2015) Green internet of things for smart world. IEEE access 3:2151–2162
9. Alsamhi S, Ma O, Ansari M, Meng Q (2018) Greening internet of things for smart everythings with a green-environment life: a survey and future prospects. arxiv 2018. arXiv preprint arXiv:1805.00844
10. Khan A, Mushtaq N, Faraz SH, Khan OA, Sarwar MA, Javaid N et al (2017) Genetic algorithm and earthworm optimization algorithm for energy management in smart grid. In: International conference on P2P, parallel, grid, cloud and internet computing. Springer, pp 447–459
11. Tiwari R, Kumar N (2015) Minimizing query delay using cooperation in IVANET. Procedia Comput Sci 57:84–90
12. Kumar S, Tiwari R (2021) Dynamic popularity window and distance-based efficient caching for fast content delivery applications in CCN. Eng Sci Technol Int J
13. Kumar S, Tiwari R (2020) Optimized content centric networking for future internet: dynamic popularity window based caching scheme. Comput Netw 179:107434
14. Kumar S, Tiwari R (2020) An efficient content placement scheme based on normalized node degree in content centric networking. Cluster Comput 1–15
15. Toor A, Islam SU, Sohail N, Akhunzada A, Boudjadar J, Khattak HA, Din IU, Rodrigues JJ (2019) Energy and performance aware fog computing: a case of DVFS and green renewable energy. Future Gener Comput Syst 101:1112–1121
16. Jalali F, Vishwanath A, De Hoog J, Suits F (2016) Interconnecting fog computing and microgrids for greening IoT. In: 2016 IEEE innovative smart grid technologies-Asia (ISGT-Asia). IEEE, pp 693–698
17. Wan J, Chen B, Wang S, Xia M, Li D, Liu C (2018) Fog computing for energy-aware load balancing and scheduling in smart factory. IEEE Trans Industr Inf 14(10):4548–4556
18. Alzeyadi A, Farzaneh N (2019) A novel energy-aware scheduling and load balancing technique based on fog computing. In: 2019 9th International conference on computer and knowledge engineering (ICCKE). IEEE, pp 104–109
19. Oma R, Nakamura S, Duolikun D, Enokido T, Takizawa M (2018) An energy-efficient model for fog computing in the internet of things (IoT). Internet Things 1:14–26
20. Luo J, Yin L, Hu J, Wang C, Liu X, Fan X, Luo H (2019) Container based fog computing architecture and energy-balancing scheduling algorithm for energy IoT. Futur Gener Comput Syst 97:50–60
21. Mahmud R, Kotagiri R, Buyya R (2018) Fog computing: a taxonomy, survey and future directions. Internet of everything. Springer, pp 103–130

22. Naranjo PGV, Pooranian Z, Shojafar M, Conti M, Buyya R (2019) FOCAN: a fog-supported smart city network architecture for management of applications in the internet of everything environments. J Parallel Distrib Comput 132:274–283
23. Sharma I, Tiwari R, Anand A (2017) Open source big data analytics technique. In: Proceedings of the international conference on data engineering and communication technology. Springer, pp 593–602
24. Tiwari R, Sharma HK, Upadhyay S, Sachan S, Sharma A (2019) Automated parking system-cloud and IoT based technique. Int J Eng Adv Technol (IJEAT) 8(4C):116–123
25. Da Xu L, He W, Li S (2014) Internet of things in industries: a survey. IEEE Trans Industr Inf 10(4):2233–2243
26. Mohammed MA, Mohammed IA, Hasan RA, Ţăpuş N, Ali AH, Hammood OA (2019) Green energy sources: issues and challenges. In: 2019 18th RoEduNet conference: networking in education and research (RoEduNet). IEEE, pp 1–8
27. Rawashdeh S, Eyadat W, Magableh A, Mardini W, Yasin MB (2019) Sustainable smart world. In: 2019 10th International conference on information and communication systems (ICICS). IEEE, pp 217–223
28. Qu Z, Wang Y, Sun L, Peng D, Li Z (2020) Study QoS optimization and energy saving techniques in cloud, fog, edge, and IoT. Complexity 2020
29. Kenyon C, Cheliotis G (2004) Grid resource commercialization: economic engineering and delivery scenarios, grid resource management: state of the art and future trends
30. Schlagkamp S, Da Silva RF, Deelman E, Schwiegelshohn U (2016) Understanding user behavior: from HPC to HTC. Procedia Comput Sci 80:2241–2245
31. Kendrick P, Baker T, Maamar Z, Hussain A, Buyya R, AlJumeily D (2018) An efficient multi-cloud service composition using a distributed multiagent-based, memory-driven approach. IEEE Trans Sustain Comput
32. Mishra SK, Puthal D, Sahoo B, Jena SK, Obaidat MS (2018) An adaptive task allocation technique for green cloud computing. J Supercomput 74(1):370–385
33. Shah AS, Nasir H, Fayaz M, Lajis A, Shah A (2019) A review on energy consumption optimization techniques in IoT based smart building environments. Information 10(3):108
34. Dighriri M, Alfoudi ASD, Lee GM, Baker T, Pereira R (2018) Resource allocation scheme in 5g network slices. In: 2018 32nd International conference on advanced information networking and applications workshops (WAINA). IEEE, pp 275–280
35. Sun Y, Lin F, Xu H (2018) Multi-objective optimization of resource scheduling in fog computing using an improved NSGA-II. Wireless Pers Commun 102(2):1369–1385
36. Estevez C, Wu J (2015) Recent advances in green internet of things. In: 2015 7th IEEE Latin-American conference on communications (LATINCOM). IEEE, pp 1–5
37. Fuselli D, De Angelis F, Boaro M, Squartini S, Wei Q, Liu D, Piazza F (2013) Action dependent heuristic dynamic programming for home energy resource scheduling. Int J Electr Power Energy Syst 48:148–160
38. Longe OM, Ouahada K, Rimer S, Ferreira HC (2015) Optimization of energy expenditure in smart homes under time-of-use pricing. In: 2015 IEEE innovative smart grid technologies-Asia (ISGT ASIA). IEEE, pp 1–6
39. Akasiadis C, Panagidi K, Panagiotou N, Sernani P, Morton A, Vetsikas IA, Mavrouli L, Goutsias K (2015) Incentives for rescheduling residential electricity consumption to promote renewable energy usage. In: 2015 SAI Intelligent systems conference (IntelliSys). IEEE, pp 328–337
40. Awais M, Javaid N, Aurangzeb K, Haider SI, Khan ZA, Mahmood D (2018) Towards effective and efficient energy management of single home and a smart community exploiting heuristic optimization algorithms with critical peak and real-time pricing tariffs in smart grids. Energies 11(11):3125
41. Ahmad A, Khan A, Javaid N, Hussain HM, Abdul W, Almogren A, Alamri A, Niaz IA (2017) An optimized home energy management system with integrated renewable energy and storage resources. Energies 10(4):549
42. Manzoor A, Ahmed F, Judge MA, Ahmed A, Tahir MAUH, Khan ZA, Qasim U, Javaid N (2017) User comfort oriented residential power scheduling in smart homes. In: International

conference on innovative mobile and internet services in ubiquitous computing. Springer, pp 171–180
43. Bharathi C, Rekha D, Vijayakumar V (2017) Genetic algorithm based demand side management for smart grid. Wireless Pers Commun 93(2):481–502
44. Ali M, Abid S, Ghafar A, Ayub N, Arshad H, Khan S, Javaid N (2017) Earth worm optimization for home energy management system in smart grid. In: International conference on broadband and wireless computing, communication and applications. Springer, pp 583–596
45. Talha M, Saeed MS, Mohiuddin G, Ahmad M, Nazar MJ, Javaid N (2017) Energy optimization in home energy management system using artificial fish swarm algorithm and genetic algorithm. In: International conference on intelligent networking and collaborative systems. Springer, pp 203–213
46. Rasheed MB, Javaid N, Ahmad A, Jamil M, Khan ZA, Qasim U, Alrajeh N (2016) Energy optimization in smart homes using customer preference and dynamic pricing. Energies 9(8):593
47. Javaid N, Ahmed A, Iqbal S, Ashraf M (2018) Day ahead real time pricing and critical peak pricing based power scheduling for smart homes with different duty cycles. Energies 11(6):1464
48. Lefort A, Bourdais R, Ansanay-Alex G, Guéguen H (2013) Hierarchical control method applied to energy management of a residential house. Energy Build 64:53–61
49. Hussain HM, Javaid N, Iqbal S, Hasan QU, Aurangzeb K, Alhussein M (2018) An efficient demand side management system with a new optimized home energy management controller in smart grid. Energies 11(1):190
50. Mohsenian-Rad A-H, Wong VW, Jatskevich J, Schober R (2010) Optimal and autonomous incentive-based energy consumption scheduling algorithm for smart grid. Innovative Smart Grid Technol (ISGT) 2010:1–6
51. Bitam S, Zeadally S, Mellouk A (2018) Fog computing job scheduling optimization based on bees swarm. Enterp Inf Syst 12(4):373–397
52. Avci M, Erkoc M, Rahmani A, Asfour S (2013) Model predictive HVAC load control in buildings using real-time electricity pricing. Energy Build 60:199–209
53. Javaid N, Ahmed F, Ullah I, Abid S, Abdul W, Alamri A, Almogren AS (2017) Towards cost and comfort based hybrid optimization for residential load scheduling in a smart grid. Energies 10(10):1546
54. Aslam S, Iqbal Z, Javaid N, Khan ZA, Aurangzeb K, Haider SI (2017) Towards efficient energy management of smart buildings exploiting heuristic optimization with real time and critical peak pricing schemes. Energies 10(12):2065
55. Samuel O, Javaid S, Javaid N, Ahmed SH, Afzal MK, Ishmanov F (2018) An efficient power scheduling in smart homes using Jaya based optimization with time-of-use and critical peak pricing schemes. Energies 11(11):3155

DoSP: A Deadline-Aware Dynamic Service Placement Algorithm for Workflow-Oriented IoT Applications in Fog-Cloud Computing Environments

Meeniga Sriraghavendra, Priyanka Chawla, Huaming Wu, Sukhpal Singh Gill, and Rajkumar Buyya

Abstract The next generation Internet of Things (IoT) applications are offering multiple services and run in a distributed heterogeneous environment. In such applications, Quality of Service (QoS) requirements are in jeopardy when the computing operations are only outsourced to the public cloud. For IoT applications, a comprehensive framework that supports QoS-aware service placement in a fog computing environment is highly required. It is a challenging task to orchestrate the time critical IoT applications in the fog environment. To alleviate this problem, this paper proposes a novel multitier fog computing architecture called Deadline-oriented Service Placement (DoSP) that provides the services both in fog and cloud nodes. This research work proposed a methodology to utilize low-cost fog resources while ensuring that the response time satisfies a given time constraint. It uses the Genetic Algorithm (GA) to dynamically determine the service placement in the fog environment. In this work, we used the iFogSim simulator to model DoSP and measured the impact of the service placement technique in terms of service deadline. It has been observed that through the proposed solution, there is a reduction in service execution delay, i.e., approximately 10.19% of the overall response time to the EdgeWard and 2.58% to the Cloud Only.

Keywords Fog computing · IoT · Cloud computing · Edge computing · Workflow

M. Sriraghavendra · P. Chawla (✉)
School of Computer Science, Lovely Professional University, Phagwara, Punjab, India

H. Wu
Center for Applied Mathematics, Tianjin University, Tianjin 300072, PR China
e-mail: whming@tju.edu.cn

S. S. Gill
School of Electronic Engineering and Computer Science, Queen Mary University of London, Mile End, UK
e-mail: s.s.gill@qmul.ac.uk

R. Buyya
Cloud Computing and Distributed Systems (CLOUDS), Laboratory, School of Computing and Information Systems, The University of Melbourne, Melbourne, Australia
e-mail: rbuyya@unimelb.edu.au

R. Tiwari et al. (eds.), *Energy Conservation Solutions for Fog-Edge Computing Paradigms*, Lecture Notes on Data Engineering and Communications Technologies 74,
https://doi.org/10.1007/978-981-16-3448-2_2

1 Introduction

Cloud computing has been facilitating individuals and organizations by extending access to remote computing resources on subscription basis. In cloud computing, the servers are located in a remote place and provide computing resources on demand. Servers being in the remote place leads to problems such as high latency, high bandwidth and energy consumption. Because of these disadvantages cloud computing is not ideal for Internet of Things (IoT) applications [1], which are latency sensitive, such as precision agriculture, intelligent transportation, smart homes, smart cities, smart grids, smart healthcare units and smart supermarkets. Providing services in a short time is a major research issue that has been addressed by many researchers in the past few years [2–4]. The issue to get the computation closer to the end devices, which produce the computational data, is addressed by fog computing technology. Fog computing is an elongated version of cloud computing with effective network bandwidth utilization, heterogeneous computation, workload distribution, and mobility. Some of the extended features of fog computing are described in Table 1.

Resource provisioning can reduce latency [36] and increase compliance of Service Level Agreements (SLAs) [5], which has been investigated in [6–10]. However, mere service provisioning cannot achieve the desired performance, which needs to be coupled with service placement strategies as explored in [2, 11–17]. So, there is a need for a comprehensive approach that not only takes care of service placement but also enhances Quality of Service (QoS) in the heterogeneous and dynamic fog-cloud computing environment.

The major contributions of this proposed work are as follows: The proposed work provides a comprehensive solution of QoS-aware service placement to realize highly optimized service placement of sequential IoT applications in a fog-cloud computing environment.

1. A Deadline-oriented Service Placement (DoSP) algorithm is proposed, which analyzes the response time of service placement indifferent layers, and determines decisions on placing modules/services of workflow-based IoT application in different layers of fog-cloud architecture.

Table 1 Fog computing features

Factor	Description
Low latency	The best facilities on the edge of the network are provided
Mobility	Disassociates the host identity to the location identity
Real-time interactions	Uninterrupted speedy service
Interoperability	Objects collaborate and communicate with each other during the transmission
Low energy consumption	Reducing computation and communication energy consumption

2. The fog-cloud environment is simulated using iFogSim toolkit and the performance of the service placement algorithm is evaluated. We are able to visualize the service placement and the different layers determine decisions on placing modules/services of multiple IoT applications in different layers of fog-cloud architecture.

The rest of the sections in the paper are organized in the following way: Sect. 2 presents a motivating scenario for the research carried out. Section 3 reviews the literature on developing and deploying IoT applications, resource allocation and application/service placement in the fog-cloud environments. Section 4 presents the system model which is the basis of the research. The complete methodology used to realize an approach for deadline-aware efficient service placement is provided in Sect. 5. Section 6 presents the study on simulation that is made with iFogSim simulator. The performance evaluation of our approach and comparison with the state-of-the-art methods is presented. Section 7 concludes the paper along with directions for future work.

2 Motivation Scenario

Fog computing helps benefit IoT devices and applications. This is an important proposition based on which this motivating scenario is conceived. Workflow-based IoT applications in the real-world need to desire latency where the edge of the network with storage and computing services (fog computing) plays a crucial role. Extending the cloud computing capability to the edge of the network, and providing a required inductive usage is by Fog Computing. Placement of IoT applications with proper utilization of cloud and fog computing resources is a game-changing approach that will have a huge impact on the deployed applications. Figure 1 shows the motivating scenario that can be used to investigate the proposition aforementioned.

As presented in Fig. 1, it is evident that a patient respiratory management system is considered as a motivation scenario. Wearable sensors such as sensors are used to detect the vital signs of a patient in relation to the cardiovascular system. The wearable devices equipped with sensors are connected to other digital infrastructure. The sensors provide heartbeat rate, changes in blood volume of the skin and oxygen level carried in both. When sensors provide redundant and irrelevant data, they need to be pre-processed with filtering techniques prior to data analytics. The results of the analysis provide the severity of the problems and accordingly further steps are taken. Sometimes the results reflect an emergency situation where immediate attention is essential. As the application is crucial, in the healthcare domain it is essential to optimize service placement for better latency besides adhering to deadlines if any. It is therefore necessary to evaluate the device modules and make decisions on placements in various layers. Low latency reflects high performance while high latency reflects low performance. This scenario motivates further research on optimal service placement as the availability of the application is indispensable. Every application

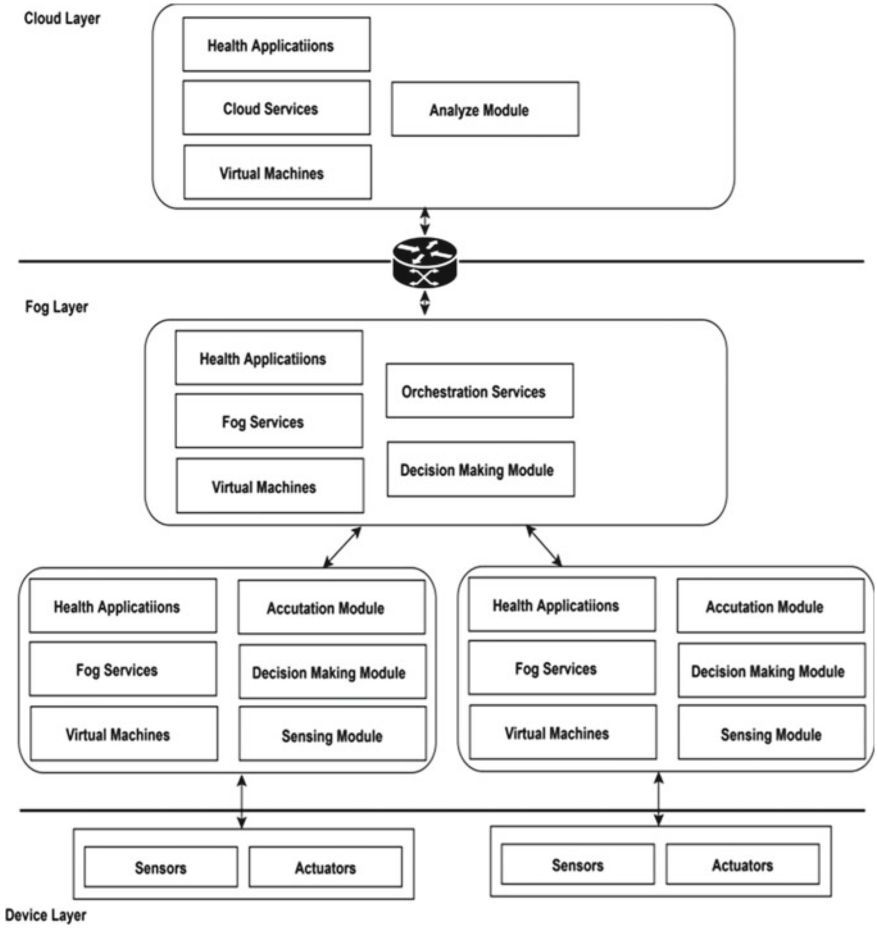


Fig. 1 Optimized application module placement architecture

consists of various modules. The modules of the case study application considered here are described in the following subsections.

2.1 Sensing Module

This is the module where actual sensing of data takes place. It is made by the wearable body sensors that are associated with a selected patient. This module is responsible for generating the patient's vital signs related to the respiratory system. Thus, it plays a crucial role in the application, because without this module, other modules would not exist.

2.2 Data Aggregation Module

This module is responsible for data aggregation, which is needed in order to have more meaningful data and also get rid of duplications. It also has a device that converts the analog signal into digital counterparts and needs to be integrated with a micro-controller unit that involves in the data aggregation process, it may include filtration and normalization. The data is aggregated and finally sent to the fog controller node.

2.3 Data Analysis Module

This module is responsible for analyzing data. This is made as per the kind of sensor data, which involves in machine learning algorithms. It is used to test hypotheses. Besides, it needs processing power, storage, and to be placed carefully based on the deadline constraints.

2.4 Decision Making

The analysis module provides the required patterns or interesting facts found in the analysis. These facts are interpreted, and decisions are made by this module. The decisions are pertaining to the real-time diagnosis of the patient's vital signs and make the necessary steps. A decision vector holds the resultant information.

2.5 Actuation Module

This module is responsible for turning energy into motion or performing its intended purpose. In general, it converts physical parameters into electrical outputs.

3 Related Work

This segment examines the literature on the location of services in the fog computing world and its associated aspects.

3.1 Building and Deployed Fog-Based IoT Applications

Kochovski and Stankovski [1] focused on dependable edge computing and proposed architecture for constructing smart applications that exploit edge computing in the context of IoT. Their proposed infrastructure includes edge management services, virtual clusters, and smart IoT construction environments. Edge management services perform probing, monitoring, alarming and storing. Edge management services work on top of virtual clusters to support edge computing applications. Pham and Huh [18] considered a cloud-fog computing environment, where fog nodes are used appropriately along with cloud nodes to enhance performance. Giang et al. [19] proposed a distributed data flow approach for building IoT applications to be deployed in fog.

Read later...

3.2 Resource Allocation for IoT Applications in Fog Environment

It is essential to have resource allocating mechanisms, when the IoT applications are executed in the fog-cloud environment. Skarlat et al. [6] presented an architecture or framework for resource allocation in the fog-cloud environment. Delay sensitive utilization of fog resources was the aim of the framework. Considering SLAs and QoS, Aazam et al. [7] proposed a framework known as Media Fog Resource Estimation for estimation of resources based on QoS, QoE, Relinquish Rate (RR) and service give-up ratio.

Yousefpour et al. [20] studied the problem of service provisioning in a dynamic and distributed computing environment. To solve the dynamic service placement they have proposed two heuristic algorithms in fog in spite of building a framework to realize it. They found that the framework could decrease SLA violations besides increasing the performance of deployed applications. Aazam and Huh [21] considered many issues with IoT-fog environment due to the mobility of nodes, resource constrained nature of nodes and heterogeneity of the environment. Taking those factors into account, they proposed resource estimation and provisioning framework through fog micro-datacenter. Aazam and Huh [22] have researched adaptive resource estimation and cost allocation models for IoT applications implemented in the fog computing area.

An architecture for flexible and adaptive resource allocation in fog computing is proposed by Agarwal et al. [8]. They proposed an algorithm known as Efficient Resource Allocation (ERA) for estimation of resources and allocating the same to IoT applications deployed in fog. On the other hand, Verma et al. [9] focused on reliable services in the fog by proposing load balancing and data replication techniques and reduce dependency on the cloud. Alsaffar et al. [23] suggested an architecture for the dual purpose of the distribution of resources and the allocation of IoT services in a fog computing system. They found that their work is resulted in meeting the QoS of applications and SLAs.

In the context of container-based service computing, Tao et al. [24] proposed a dynamic resource allocation algorithm, which is based on scheduling and fuzzy inference system. Their contributions may increase the efficiency of the cluster in terms of average execution time. Ni et al. [25] proposed a model known as Priced Timed Petri Nets (PTPN), which is a resource allocation strategy in fog computing and also a model to support the strategy. Shekhar et al. [10] proposed a dynamic and data-driven cloud and edge system for dynamic resource management in fog in the presence of performance-sensitive applications. Dang and Hoang [26] proposed a framework for efficient scheduling of tasks in the fog to realize expected latency, this framework is known as fog-based region. A joint optimization approach for resource allocation is investigated by Zhang et al. [27] for efficient resource allocation and performance of IoT applications in fog-cloud environments.

3.3 Service Placement in Fog Environment

IoT applications may have micro-services that need to be deployed strategically. Filip et al. [11] studied cloud-edge environments for scheduling micro-services related to IoT applications. They proposed a scheduling architecture for real-life utilization of the fog-edge environment effectively. They used a scheduling policy for better task scheduling. IoT applications have multiple modules. Mahmud et al. [2] studied deploying such modules in the fog-cloud environment. They proposed a methodology to fully take advantage of the capabilities of fog nodes in the context of large IoT applications. Their approach resulted in a decrease of latency toward QoS requirements of the applications. Similarly, Desikan et al. [28] explored latency-aware data processing in the fog-cloud environment. They could reduce the response time and increase the performance of gateways with buffer occupancy efficiencies.

Taneja and Davy [12] proposed an algorithm named Module Mapping, which is meant for deploying modules of IoT application appropriately in fog-cloud resources. As the storage and computation are distributed dynamically, the algorithm was able to reduce latency and improve the capabilities of the cloud-fog environment to withstand SLAs. Skarlat et al. [29] investigated QoS-aware provisioning of services of IoT, on fog resources and found a considerable 35% less cost in terms of execution time while compared with the cloud-based approach. On the other hand, Mahmud et al. [30] considered Quality of Everything (QoE)-aware placement of services or applications in a fog computing environment. They proposed an application policy that is QoE-aware and prioritizes application placement requests as per the expectations of users.

Souza et al. [13] studied the QoS-aware service allocation problem. They proposed fog-cloud architecture with four tiers. The bottom layer composed of end-user devices, the layer above is called the fog layer 1, fog layer 2 is on top of fog layer 1 and the top most layer is the cloud. The cloud has high reach ability but causes high access delay with respect to IoT applications. Fog layer 2 exhibits medium reach ability and medium access delay, while fog layer 1 causes low reach ability and low delay. This scenario is to be exploited by QoS-aware service allocation. FogTorch is a java tool which is proposed by Brogi and Forti [14]. The tool is a latency and bandwidth-aware while deploying the IoT applications in the fog. Brogi et al. [31] have reviewed the existing policies to solve the problem of how to dynamically deploy the application modules in the computational components with qualitative attributes. Abbasi et al. [32] address the problem of placement of workloads. The purpose of their study is to minimize the energy consumption and propagation delay of the cloud while processing. In this work, the placement problem is solved by using the NSGA-II algorithm. The work does not focus on the application deadline and the propagation delay among the fog nodes.

3.4 A Qualitative Comparison

Table 2 identifies and compares key elements of related works with DoSP. The comparison attributes are the target system, application type, resource utilization, minimization of response time, application priority and deadline constraint. The insights found in the literature reveal that the existing approaches are good to realize fog-cloud environments with increased performance. However, it is understood that a comprehensive framework that not only takes care of service placement but also QoS enhancement in the presence of the highly heterogeneous environment.

4 System Model

The system model considered for the proposed research provides the layered architecture (Fig 2), in which proposed workplaces the multiple sequential IoT applications with a sense-process-actuate model are executed. In the context of this system model service placement, QoS-aware service placement approach is explored in this research work.

FN is meant for storing arbitrary data required by workflow-based IoT applications and also supports computations. It is said to be the local node or the node that is closer to the deployed IoT application. Thus, fog computing is an enhanced version of cloud computing which is especially meant for rendering efficient services to workflow-based IoT applications. As per the facts found in the literature, FCN > FN > CN > NFCN is the thumb rule with respect to the speed of the deployed services of IoT applications. In this context, this system model is used to investigate service placement with QoS awareness strategies in place. The proposed mechanism for service placement has two different phases, i.e., application prioritizing phase and node selection phase. In the prioritizing phase, the services of IoT applications are assigned priorities. In the node selection phase, the suitable node in the fog or cloud is selected for the placement of given services. Based on the objective function, the node selection is carried out (Fig. 2).

Table 2 Comparison of DoSP with existing works

Work	Target system	App (application) type	Resource utilization (same cluster/neighbor cluster)	Minimize response time	App (application) priority	Deadline constraint	Approach
Kochovski et al. [1]	Edge	Workflow	Same cluster	No	No	No	Test bed
Giang et al. [19]	Fog	Workflow	Same cluster	No	No	No	Visual programming tool
Taneja et al. [12]	Fog	Workflow	Same cluster	No	No	No	Heuristic algorithm
Abbasi et al. [32]	Fog	Workflow	Same cluster	No	No	No	Heuristic algorithm
Skarlat et al. [6]	Fog	Workflow	Same cluster	Yes	No	No	Linear programming
Pham et al. [18]	Fog	Workflow	Same cluster	Yes	No	No	Heuristic algorithm
Alsaffar et al. [23]	Fog	Workflow	Same cluster	Yes	No	No	Heuristic algorithm
Gupta et al. [33]	Fog	Workflow	Same cluster	Yes	No	No	Heuristic algorithm
Souza et al. [13]	Fog	Workflow	Same cluster	Yes	No	No	Linear programming
Zeng et al. [34]	Fog	Bag-of-tasks	Same cluster	Yes	No	No	Heuristic algorithm
Mahmud et al. [30]	Fog	Workflow	Same cluster	Yes	Yes	No	Fuzzy logic
Goudarji et al. [17]	Fog	Workflow	Same cluster	Yes	No	No	Heuristic algorithm
Pham et al. [3]	Fog	Workflow	Same cluster	Yes	No	Yes	Heuristic algorithm

(continued)

Table 2 (continued)

Work	Target system	App (application) type	Resource utilization (same cluster/neighbor cluster)	Minimize response time	App (application) priority	Deadline constraint	Approach
Mahmud et al. [2]	Fog	Workflow	Same cluster	Yes	Yes	Yes	Heuristic algorithm
Skarlat et al. [4]	Fog	Workflow	Same and neighbor Cluster	Yes	Yes	Yes	Linear programming
DoSP (this work)	Fog	Workflow	Same and neighbor cluster	Yes	Yes	Yes	Heuristic algorithm

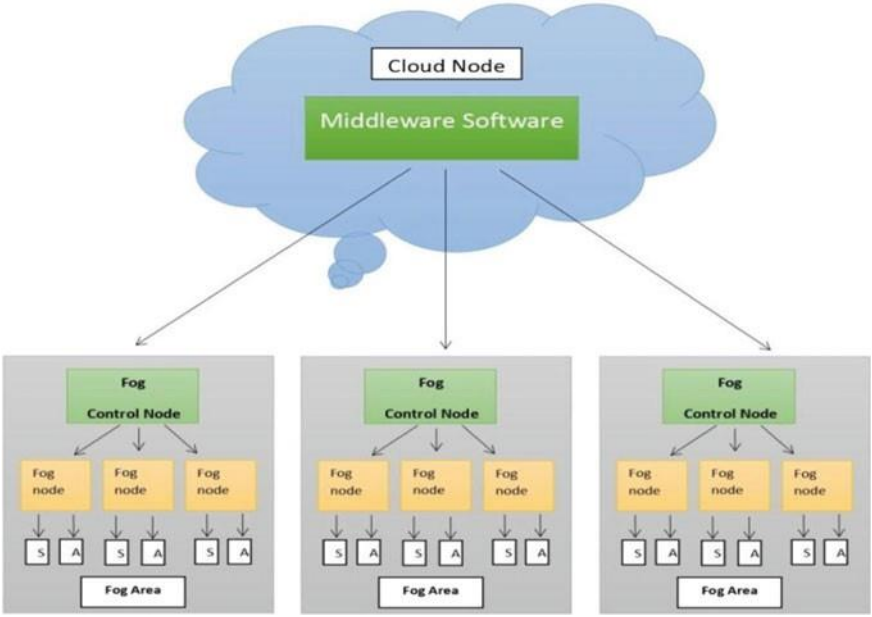


Fig. 2 System model

4.1 Application Prioritizing Phase

In this phase, the requests coming from different applications for service placement are subjected to prioritization. Toward this end, based on the distance between the deadline of an application and the deployment time of an application we prioritize the applications, which is defined as follows:

$$\text{Prioritization} = \text{Deadline}_{\text{App}_i} - \text{Deptime}_{\text{App}_i} \tag{1}$$

where $\text{Deadline}_{\text{App}_i}$ represents application deadline, and $\text{Deptime}_{\text{App}_i}$ represents waiting time for deployment. $\text{Deptime}_{\text{App}_i}$ is associated with the time waited by application prior to assignment of it to suitable computing resources.

It is always a better practice to initially assign the applications that have high waiting time based on the deadline to the fog resources. As the deadline cannot be violated, it needs to be made so. Allocating the shortest computation applications first to fog resources may lead to starvation, hence it is not desirable.

4.2 Node Selection Phase

In order to deploy services of an application, it is essential to have certain constraints that can help to achieve optimal performance.

Constraint 1: In addition to storage, this constraint is related to RAM and CPU (important computer resources). These resources are to be considered as the fog resources allocation should not exceed these available resources.

$$\sum_{App_i}^{App}\sum_{AppMod_i}^{App_i} Res_{AppMod_i} \cdot Place^{fn}_{AppMod_i} \leq Avail^{Res^{fn}}, \forall fn \tag{2}$$

where fn = {FN, FCN, NFCN} denotes fog node in a fog cluster, Res = {CPU, MEM, STORAGE} denotes required resources, Place denotes service placement and Avail denotes available resources.

Constraint 2: This condition is related to expected response time of the applications. No application should violate its deadline, therefore:

$$Resptime_{App_i} \leq Deadline_{App_i}, \forall App_i \tag{3}$$

where $Resptime_{App_i}$ represents application response time and $Deadline_{App_i}$ represents application deadline.

4.2.1 Estimating Application Response Time

In a deadline-aware approach, it is important to estimate application response time. Therefore, the response time can be estimated by:

$$Resptime_{App_i} = TotDeptime_{App_i} + Exectime_{App_i} \tag{4}$$

$$Exectime_{App_i} = Makespan_{App_i} + Comm_{App_i} \tag{5}$$

where $TotDeptime_{App_i}$ denotes deployment time, $Exectime_{App_i}$ denotes execution time, $Makespan_{App_i}$ denotes makespan time, and $Comm_{App_i}$ denotes communication time.

$Deptime_{App_i}$ considers elapsed time prior to the placement of each service correctly. The placement is done either in cloud or fog based on runtime situations. In fact, $TotDeptime_{App_i}$ represents elapsed time and also the expected deployment time. It is made when service $AppMod_i \in App_i$ is actually subjected to propagation to nearest colony. The execution time denoted as $AppMod_i \in App_i$ reflects the time required for execution of all services of given application plus the amount of communication needed among services. Therefore, the total execution time $AppMod_i \in App_i$ is computed as the communication time $Comm_{App_i}$ and the sum of makespan $Makespan_{App}$. In the link, delays reflect communication time in the process of service placement in different areas as illustrated in the architecture.

4.3 IoT Application Placement Flow in Fog Environment

Figure 3 shows a fog controller node receives a service placement request from the user. Then, it identifies the required resources for the requested IoT application. It also takes deadline associated with the service to be placed. Keeping deadline in mind, the resources on fog nodes are analyzed. Then identify fog nodes that can meet service deadline. Afterward, the deadline deployment time is calculated and verified. If it is low, the service is placed in the fog node (or) fog controller node (or) cloud. If the time is high, then it is forwarded to NFCN and checks the resource availability and QoS expectation. If it is satisfied, the application modules are placed in NFCN, otherwise, they are forwarded to the public cloud where the service is placed.

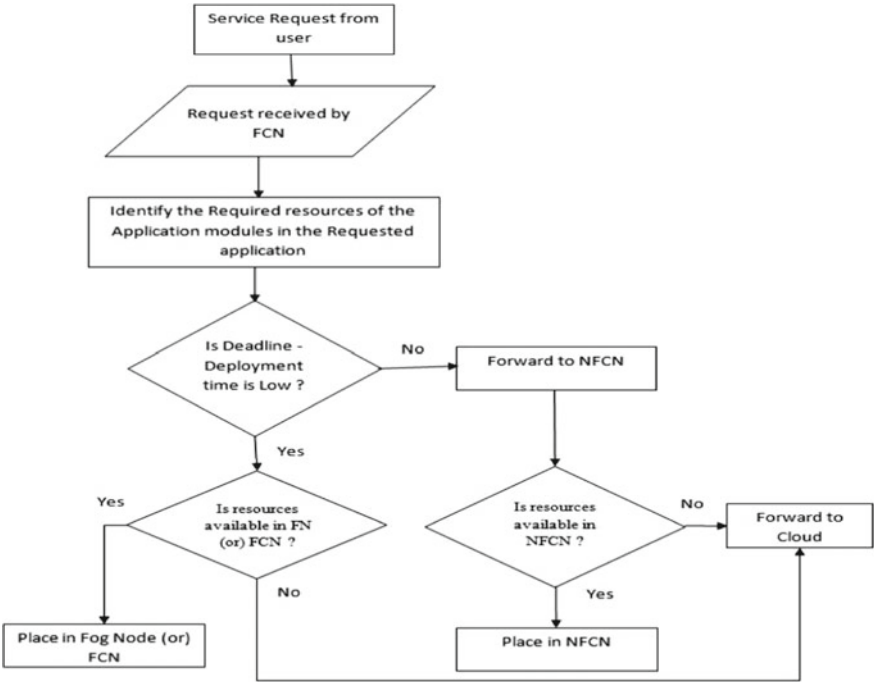


Fig. 3 Flowchart of resource provisioning for IoT application in fog-based computing environment

5 Proposed Methodology

Goal of this proposed work is to develop a comprehensive framework that supports QoS-aware service placement of IoT-based applications in a fog-based computing environment. The significance of this methodology lies in the context of emerging IoT applications in different domains like healthcare, military and the need for fog computing to accommodate the same besides exploiting the technologies like cloud computing. According to the research objectives mentioned in Sect. 4, the proposed methodology needs to be divided into multiple sections.

5.1 Overview of the Proposed Work

This section provides an overview of the proposed work. The framework presented in Fig. 4, which provides QoS-aware service placement for ensuring that the placement of services is highly optimized to improve performance. This kind of methodology can help in placing services in a fog computing environment with efficient decision making related QoS requirements of a given IoT application. Workflow-based IoT application(s) is given as input. Then, the proposed system is responsible for analyzing its requirements and executing an algorithm to have an effective service placement that considers the deadline of services. The service placement achieves satisfying deadlines. Here in the proposed work, no service layer agreements are considered, for simplicity.

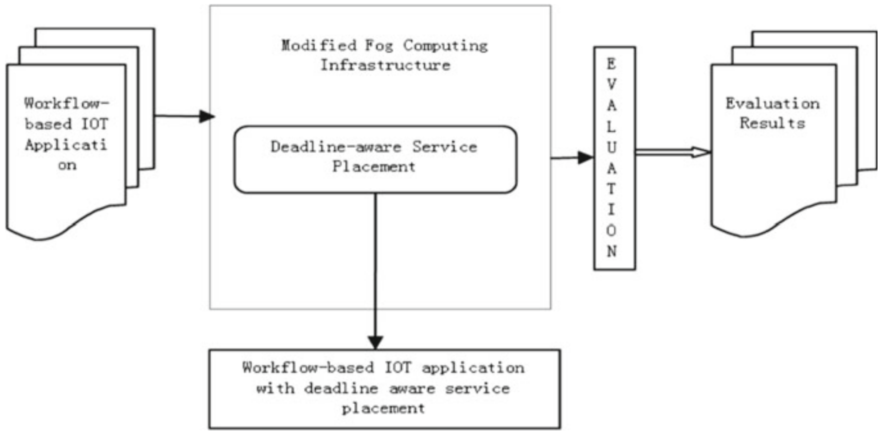


Fig. 4 Overview of the framework showing the proposed research

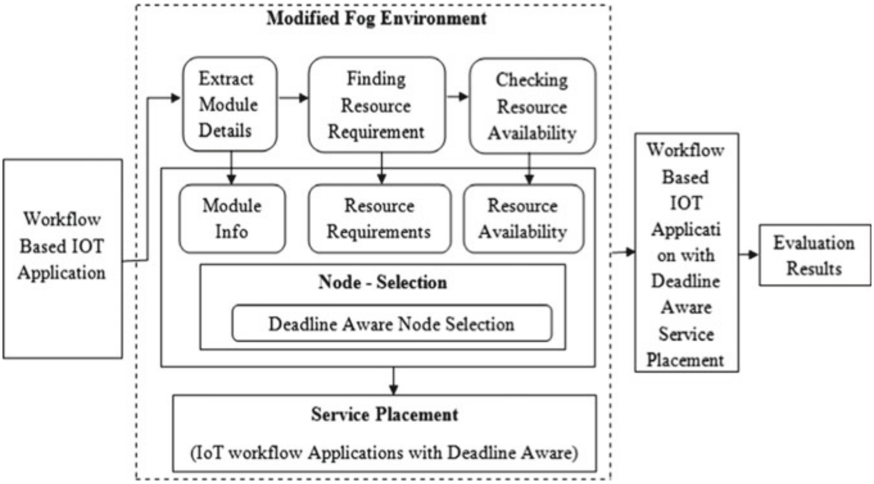


Fig. 5 Functionalities of the proposed architecture

5.2 Functional Details of the Proposed Work

The proposed architecture for deadline-aware service placement is as shown in Fig. 5, which is elaborated with functional details. On taking one or more workflow-based IoT applications, the proposed architecture performs a series of activities to ensure that the service placement is done according to the requirements of the application. From the IoT application, the modules of the application are extracted. For each module/service, the details obtained include a number of modules to be placed, the sequence of the modules and the deadline associated with each module. Afterward, the requirements for each module are analyzed. This is nontrivial as each module needs different resources such as memory, processing power, and bandwidth. Moreover, this requirement for a given service may be fulfilled in FN, FCN, NFCN or CN. Resource availability is then verified to know the information about all nodes and their available resources. The resource availability information is then utilized to make a decision on node selection for a given service or module. Once node selection is made based on the latest analysis of the resource availability considering the deadline associated with the service, the services are finally placed in a selected layer and in selected nodes.

5.3 Deadline-oriented Service Placement Algorithm (DoSP)

Deadline-oriented Service Placement (DoSP) algorithm is designed to analyze the fog environment for a given deadline requirement of a workflow-based IoT application and make well-informed decisions on the application or its modules placement in the fog environment. The proposed DoSP algorithm is presented in Table 3.

The proposed algorithm is part of deadline-aware service placement module in Fig. 4 and node selection module in Fig. 5. It is based on the concept of Genetic Algorithms (GA). When using a genetic algorithm, the first thing to be considered is how to model the solution.

Every operator inside the GA will work using that model. The size of the vector considered is equal to the number of services that will be used in the algorithm, i.e., the number of services requested to be executed. The vector is the placement

Table 3 Deadline-oriented Service Placement (DoSP) algorithm

Algorithm 1: Deadline-oriented Service Placement (DoSP)
Inputs: No. of applications, No. Application modules, CPU, Memory Storage, make-span time, deployment time, population.
Output: Efficient and deadline-aware service placement

// build the first generation
Chromosome Length = total No. of applications * No. Application modules;
Number Of Placement Locations = FN or CN or NFCN or FCN;

for *i* = 1 to *population Size* **do**
  new Chromosome = create Chromosome(*Chromosome Length*, *Number Of Placement Locations*);
   if(state_check(Chromosome))
    calculate fitness of new Chromosome;
    add new Chromosome to *population*;
    i++
end for
for *i* = 1 to *generation Limit* **do**
  fittest Chromosome = get Fittest Chromosome(*population*);
 if fitness of fittest Chromosome > 0 then
    return fittest Chromosome;
  add the current population to the temporary population;
 number Of Chromosomes Used For Crossover = generation Size * defined Crossover Percentage;
  number Of Crossovers = *number Of Chromosomes Used For Crossover* / 2;
// two chromosomes are involved in each crossover
  for *j* = 1 to number Of Crossovers **do**
    parent Chromosomes = select Chromosomes(*population*); // selection
    children Chromosomes = mate Chromosomes(*parent Chromosomes*); // crossover
    state_check(*children Chromosomes*)
   state_check(mutate Chromosomes(*children Chromosomes*)); //mutation and state check
    calculate the fitness of the new chromosomes;
    add the new chromosomes to temporary population;
  end for
  add elite chromosomes from temporary population to the new population;
  add remaining best chromosomes from temporary population to the new population;
end for

plan which places the application modules. If the vector $[i]=2$, then *i*th module is placed in node 2. Hence, the *i*th module is placed in the computation node value. We consider this vector (the placement plan) as the chromosome. As said before, it will be in the form of a vector of integers. Possible integers are IDs of fog cells, cloud, closest neighbor, fog orchestration control node. The IDs considered as follows, 0 for Cloud, 1 for FCN, 2 for NFCN, 3 to remaining as FN. The number of possible integers is assigned to the variable "number of Placement Locations".

After we assign values to variables "Chromosome Length" and "Number of Placement Locations", we need to create the first generation. One of the segments of the GA is the population size that needs to be predefined. With this number, we will go ahead and create many chromosomes (solutions) to form the first generation. There are a lot of different ways how we can create a chromosome, i.e., fill a vector of integers. One way is to randomly put numbers from a set of possible integers to the vector. A new chromosome is created at this stage. All the generated chromosomes should satisfy the state check. A state check is an operation which checks whether the given plan is successfully placed. A module can be successfully placed firstly, when the required MIPS is less than available MIPS secondly, the sense and actuation modules are placed only in FN, and finally process modules are not allowed to place in FN.

After a new chromosome has been created, we calculate its fitness to see how good it really is, and then we add it to the population. The fitness of the chromosome depends on how far the app is away from the deadline. If the application is closer to the deadline then a high penalty is added or else a low penalty is added. Here we need to minimize the fitness value. Minimum fitness chromosome is the best possible placement plan. We assign the fittest chromosome from the population to the variable "fittest Chromosome". If that chromosome satisfies all the constraints, we return it, and this is done. If that is not the case, we will need to use the rest of the algorithm. First, we define how many crossovers will be executed. "Crossover" is another operator inside the genetic algorithm which is used.

Another segment of the genetic algorithm is the number of generations, which is also user-defined. In each generation, we used a GA operator "selection" to get the best (fittest) chromosome. The fittest chromosome is, in this case, the one that has the biggest fitness among the selected ones. So, we are not looking at the fittest chromosome directly from the population. The selection part of the algorithm can be implemented in many ways. One way is to use the Tournament selection. Tournament selection is the type of selection with two steps.

The first step is to round all the solutions from the population into a "circle", and depending on the fitness of each solution, it will occupy some part of that circle. The better is the fitness, the greater the percentage of the circle. The second step is to return the fittest out of the selected solutions. This selection is used to get both parents. There are different ways of how a crossover can be implemented.

One way is the uniform crossover. In this crossover, genes (parts of the chromosome) are uniformly selected between both parents to form a child chromosome. Another genetic algorithm parameter is the fixed mixing ratio inside the uniform crossover, which tells us how much of the genes will come from each parent. We are actually creating two child chromosomes with crossover. Those chromosomes are completely opposite. If the first one uses the first gene from the first parent, the second one will use the first gene from the second parent, etc. Here in the proposed algorithm, we have used uniform crossover. The resultant chromosome is the valid chromosome which satisfies the state check. A mutation can happen after crossover. The mutation is another genetic algorithm operator. It is rarely used, but it is a tool that helps a lot when we want to have a diverse population. In the mutation operation, we select "*m*" number of random locations, and we stuff each location with a new node value that satisfies the state check.

After the mutation, we calculate the fitness of the new chromosomes and store them (chromosomes) into a temporary and constant list of population. In that temporary list, we also put the whole current population. After this, we do all the crossovers and place all the new chromosomes to the temporary population, we perform elitism. Elitism has a percentage, for example, 20%, which means that 20% of the best chromosomes from the current population will automatically go to the next generation. The rest will be selected from the temporary population by sorting them (chromosomes) based on the fitness value and choosing the best 80% from them. This depends on when the user-defined algorithm stops. When the number of applications and nodes increase the time complexity will increase, this is observed while using genetic algorithms. The asymptotical time consumption or the complexity of the proposed algorithm is $O(nlogn+IPS2)$, where n indicates no. of applications for sorting, I is the no. of iterations, P indicates population size and S implies no. of sub solutions.

6 Performance Evaluation

6.1 System Setup and Parameters

As found in [7, 11, 12, 29], simulation studies are widely used for fog computing to evaluate resource management policies. Here, we use a fog computing simulation toolkit named iFogSim [8], which is found to be a suitable framework for fog computing research as it is used by a number of research works like [15, 23, 27, 35].

The environment is important for conducting experiments. A summary of the environment used for the empirical study in this research work is given in Table 4.

Table 4 Simulation setup and its configuration

Processor	Intel core i5-2430 CPU, 2.40 GHz
Memory	8 GB
Simulator	iFogSim
Operating system	Windows 7 Professional
Topology model	Hierarchy

The simulation study has been carried out using iFogSim with JDK 1.8. Our simulation framework has an API to visually model the fog computing infrastructure. It has provisions to model the cloud, fog nodes, fog controller nodes, gateways, sensors, and actuators. The organization of the network is as follows: one "Fog Orchestration node", ten "Fog Nodes" controlled by a "Fog Controller Node", one "Neighbor Controller Node", and a "Cloud". The processing capability of the node can be set with a certain capacity that is measured in Millions of Instructions per Second (MIPS). The iFogSim supports drag and drop features that help users to have intuition in design and understanding.

Figure 6 shows the fog infrastructure modeled for the proposed simulation study.

Each fog cell has a provision for different nodes. Once the modeling is made, the nodes are configured appropriately. In the simulation study, each node mimics a real-world node with equivalent characteristics. Therefore, it is essential to configure them as required. How fog controller and fog nodes are configured for different parameters is shown in Table 5. The parameters include processing rate, memory and storage details.

As shown in Table 6, there are other parameters needed for simulation. These parameters are configured in terms of the communication link delays (in seconds). This makes it easier to observe and measure the results in the simulation study. When

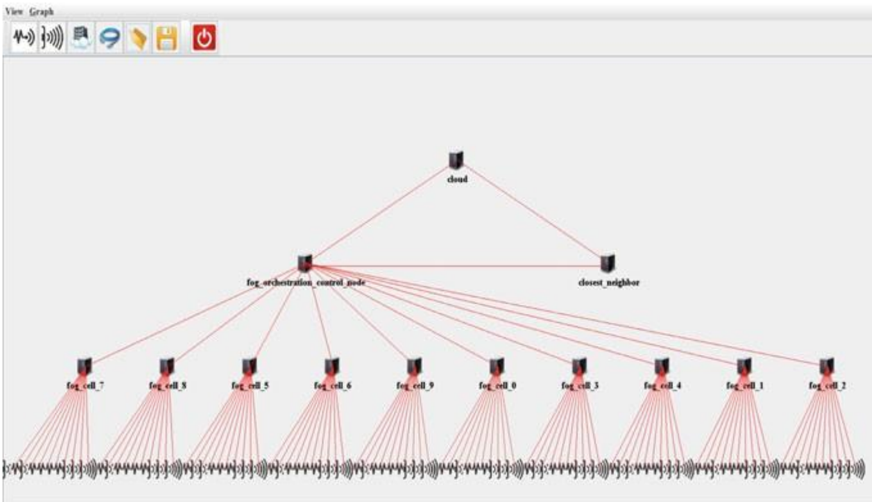


Fig. 6 Fog infrastructure modeled using iFogSim for the simulation study

Table 5 Characteristics of the fog controller node and fog node

Parameter	Fog controller node	Fog node
Processing rate (MIPS)	1000	250
Memory (MB)	512	256
Storage (GB)	8	4

Table 6 Different parameters and their corresponding communication link delay in seconds

Parameter	Communication link delay (sec)
Fog controller node-cloud	9
Fog controller node-neighbor controller node	0.5
Fog controller node-fog node	0.3

the service placement scenario is simulated, observation of various parameters is essential as those parameters are used in the real-world fog computing networks.

The communication link delay for fog controller node–fog node link is set to 0.3 s. The communication link delay for fog controller node–neighbor controller node link is set to 0.5. The communication link for fog controller node–cloud is configured to have 9 s delay.

Once such configurations have been done, it is important to take care of service modules as well. They need different resources like CPU, memory, storage, and makespan. These configurations are listed in Table 7. These details are provided for the well-informed simulation study. For instance, the actuate module is set to have 50 CPU (MIPS), 20 MB main memory and 10 MB storage and 0.50 makespan time in seconds.

The parameters, deadline and deployment, configurations for different types of applications are as shown in Table 8.

For application types associated with motion, video, sound, temp, and humidity, the application deadline and deployment time are determined and presented. The

Table 7 Required resources for application modules

Service module	CPU (MIPS)	Memory (MB)	Storage (MB)	Makespan (sec)
Sensing module	50	30	10	0.90
Data aggregation module	200	10	30	0.10
Data analysis module	200	20	30	0.10
Decision making module	100	30	30	0.25
Actuate module	50	20	10	0.50

Table 8 Deadline and deployment time of each application

Application type	Application deadline	Deployment time
Motion	120	60
Video	300	0
Sound	300	60
Temp	360	60
Humidity	240	0

values that were assumed here in Tables 6, 7, and 8 are based on the average of the previous experimental run.

6.2 Experimental Results

Observations are made on application or its modules placement in fog computing using the experimental setup described in the previous section. The proposed deadline-aware method for service placement methodology is compared with the contemporary methods, e.g., EdgeWard [33] and Cloud Only. EdgeWard is a baseline greedy optimization heuristic, hence it is chosen. EdgeWard places the application modules in the bottom to top approach, based on the availability of resources. In all the cases, Cloud Only places all the application modules in the cloud. Response time is an important observation made at different deadlines. Different workflow-based IoT applications are considered for empirical study. Simulations using iFogSim have resulted in desired insights for the applications such as motion, video, sound, temp, and humidity.

As depicted in Fig. 7, the deadline requirement for Motion Application is 120 s and Video Application is 300. The horizontal axis provides different service placement approaches, while the vertical axis shows response time in seconds. The results revealed that deadline violation occurred in the case of motion application. In both cases of EdgeWard and Cloud Only methods, the deadline is violated by 184.45 and 31.85 s, respectively. The *EdgeWard, Cloud Only and DoSP* approaches could ensure that the time limit for video application is not breached.

As can be seen in Fig. 8, it is evident that the results are provided for sound and temp applications. The response time is observed for the IoT applications with deadlines 300 s and 360 s, respectively. The results revealed that deadline violations occurred in the case of sound application. The EdgeWard method could not provide service placement ideally as it violated the deadline by 4.45 s. In the case of temp application, service placement is made by the three methods without violating the deadline. However, EdgeWard took more response time while DoSP method took the least response time.

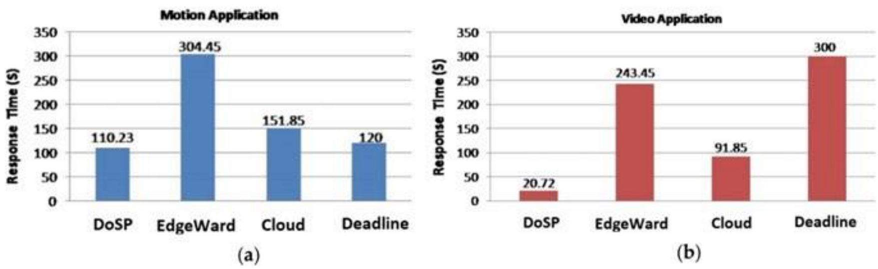


Fig. 7 Performance comparison with **a** motion, and **b** video applications

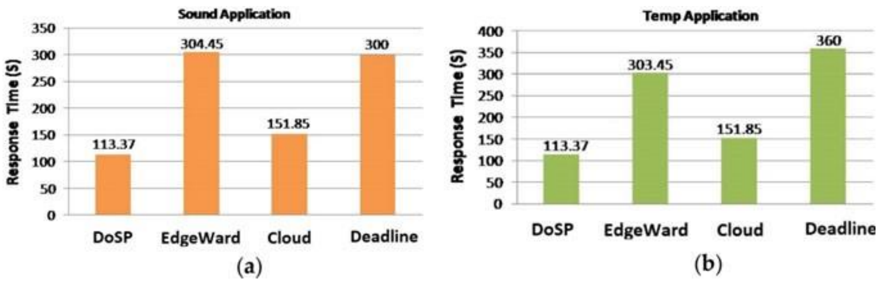
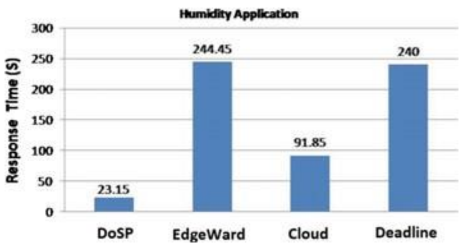


Fig. 8 Performance comparison with **a** sound, and **b** temp applications

Fig. 9 Performance comparison with humidity application



As presented in Fig. 9, the performance of service placement approaches showed different response times. The proposed DoSP method could honor the deadline. It is the same with Cloud Only approach as well. In the case of EdgeWard method, the deadline is violated by 4.45 s. This kind of performance is not acceptable as deadlines in cloud and fog computing scenarios might be associated with SLAs.

6.2.1 Response Time Against Deadline

Response time or latency is an important metric used for evaluating the proposed DoSP algorithm. The response time against a given deadline is an important empirical observation. As shown in Fig. 10, the observations are made in terms of response time against different deadlines. Results of all the applications are presented with the communication distance between the fog controller node and the cloud is 1 s. The deadlines from 50 to 500 s incremented by 50 s are provided on the horizontal axis.

The response time is shown on the vertical axis. When the elapsed time is 50 s, both Humidity and Video applications started while other applications started at an elapsed time of 100 s. The observations revealed that the response time values can be categorized into 3 levels, namely, lower, medium and high. When services are placed in a fog controller node, response time is less indicating higher performance. In the same fashion, when services are placed in the cloud, the response time is said to be medium, which is greater than that of the fog controller node. The response time is more when services are deployed in the neighbor fog controller node due to

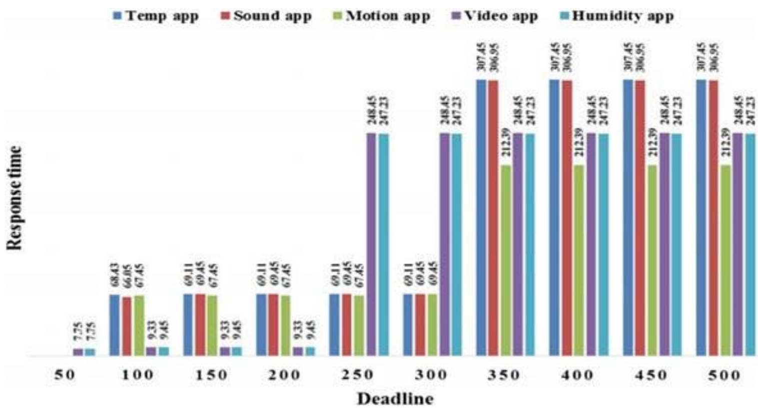


Fig. 10 Performance of the DoSP method with respect to all applications

deployment delay and extra deployment time to place the service modules. These observations provide the rationale behind the patterns found in response time. The results revealed that the deadline has an impact on the response time. The rationale behind this is that the purpose of the proposed DoSP algorithm is to provide service placement which yields response time that should not violate the given deadline. The results revealed that the proposed method is deadline-aware, and no violations are found with respect to all applications. It does mean that based on the deadline and the resource availability, DoSP has taken appropriate service placement decisions.

6.2.2 Resource Utilization Analysis

Resource availability is analyzed by the proposed DoSP method in order to make service placement decisions. In the EdgeWard method, the sensing and actuating services are placed in various fog nodes in the fog layer. Other activities are known as data aggregation, analysis of data and decision making, are moved to the controller node and propagated to the neighbor fog controller node. In the DoSP, it is different. The sensing and actuation modules are placed in the fog nodes whereas the processing-related modules are placed in either fog controller node or in the cloud node.

As shown in Fig. 11, based on the resource analysis, 40% of the service placements are made in fog node, which is faster than other nodes. Service placement shows 12% in the closest neighbor node and 6% in the fog controller node.

The remaining 42% of service placements occur in the cloud. As mentioned earlier, there is a difference in EdgeWard scenarios. It places 40% of services in the fog nodes, 32% in the closest neighbor nodes and 28% in the fog orchestration control nodes.

The Cloud Only approach keeps everything in the cloud. Therefore, it is not able to exploit the fog nodes and fog controller nodes. It has a drawback as the Cloud

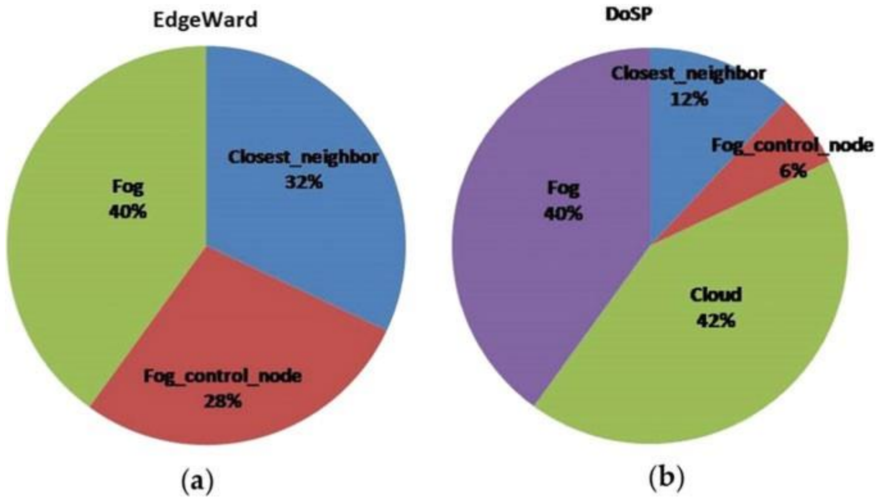


Fig. 11 Resource utilization for the **a** EdgeWard method, and **b** DoSP

Only service placement may not meet certain workflow-based IoT applications where quick response time is desired. Cloud is relatively slower than other layers in the fog computing architecture.

7 Conclusion and Future Work

We proposed a Deadline-Aware Dynamic Service Placement (DoSP) algorithm for multiple sequential IoT workflow applications of the same model (sense-process-actuate) and the same number of modules in each application. The applications follow the DDF deployment model. Each application contains the same number of services and each and every service has to be placed on computational nodes.

With the advent of fog computing a platform for the exploitation of available resources at the edge of the network is emerged. The response time of different layers can be stated as Neighbor fog controller node $(\text{NFCN}) > \text{Cloud node } (\text{CN}) >$ Fog node $(\text{FN}) >$ Fog controller node (FCN). The proposition is interesting as it can speed up IoT services deployed in fog computing. The proposed work investigates the proposition of the hypothesis aforementioned. It has architecture with multiple layers, i.e., device layer, fog layer, and cloud layer. An evaluation of DoSP algorithm is carried out using iFogSim simulator. The simulation results revealed that the proposed algorithm offers better performance over the approaches like EdgeWard, Cloud Only. DoSP performed well as its placement strategy differs to optimize performance in terms of reducing response time in the presence of strict deadlines associated with SLAs. This conclusion is made using only five workflows based IoT applications but this may not be limited to five and can be extended. The values that were assumed are based on the average of the previous experimental run, which are not extracted from any real-time dataset. In the future, we plan to work on the real-time dataset and investigate energy efficiency for optimization of service placement along with the deadlines considered. In further study a hyper-heuristic approach can also be used for better optimization. Container virtualization along with device mobility in a fog computing environment would also be considered.

References

1. Kochovski P, Stankovski V (2018) Supporting smart construction with dependable edge computing infrastructures and applications. Autom Constr 85:182–192
2. Mahmud R, Ramamohanarao K, Buyya R (2018) Latency-aware application module management for fog computing environments. ACM Trans Internet Technol (TOIT) 19(1):1–21
3. Pham X-Q, Man ND, Tri NDT, Thai NQ, Huh E-N (2017) A cost-and performance-effective approach for task scheduling based on collaboration between cloud and fog computing. Int J Distrib Sens Netw 13(11):1550147717742073
4. Skarlat O, Nardelli M, Schulte S, Dustdar S (2017) Towards QoS-aware fog service placement. In: 1st IEEE international conference on fog and edge computing (ICFEC) IEEE, Madrid, Spain, pp 89–96
5. Wu H, Knottenbelt WJ, Wolter K (2019) An efficient application partitioning algorithm in mobile environments. IEEE Trans Parallel Distrib Syst 30(7):1464–1480
6. Skarlat O, Schulte S, Borkowski M, Leitner P (2016) Resource provisioning for IoT services in the fog. In: 2016 IEEE 9th international conference on service-oriented computing and applications (SOCA). IEEE, pp 32–39
7. Aazam M, St-Hilaire M, Lung C-H, Lambadaris I (2016) MeFoRE: QoE based resource estimation at fog to enhance QoS in IoT. In: 2016 23rd International conference on telecommunications (ICT). IEEE, pp 1–5
8. Agarwal S, Yadav S, Yadav AK (2016) An efficient architecture and algorithm for resource provisioning in fog computing. Int J Inf Eng Electron Bus 8(1):48
9. Verma S, Yadav AK, Motwani D, Raw R, Singh HK (2016) 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
10. Shekhar S, Gokhale A (2017) Dynamic resource management across cloud-edge resources for performance-sensitive applications. In: 2017 17th IEEE/ACM international symposium on cluster, cloud and grid computing (CCGRID). IEEE, pp 707–710
11. Filip ID, Pop F, Serbanescu C, Choi C (2018) Micro services scheduling model over heterogeneous cloud-edge environments as support for IoT applications. IEEE Internet Things J 5(4):2672–2681
12. Taneja M, Davy A (2017) Resource aware placement of IoT application modules in fog-cloud computing paradigm. In 2017 IFIP/IEEE symposium on integrated network and service management (IM). IEEE, pp 1222–1228
13. Souza VBC, Ramírez W, Masip-Bruin X, Marín-Tordera E, Ren G, Tashakor G (2016) Handling service allocation in combined fog-cloud scenarios. In: 2016 IEEE international conference on communications (ICC). IEEE, pp 1–5
14. Brogi A, Forti S (2017) QoS-aware deployment of IoT applications through the fog. IEEE Internet Things J 4(5):1185–1192
15. Gill SS, Garraghan P, Buyya R (2019) ROUTER: fog enabled cloud based intelligent resource management approach for smart home IoT devices. J Syst Softw 154:125–138

16. Wu H, Wolter K (2017) Stochastic analysis of delayed mobile offloading in heterogeneous networks. IEEE Trans Mob Comput 17(2):461–474
17. Goudarzi M, Wu H, Palaniswami M, Buyya R (2020) An application placement technique for concurrent IoT applications in edge and fog computing environments. IEEE Trans Mob Comput (TMC)
18. Pham X-Q, Huh E-N (2016) Towards task scheduling in a cloud-fog computing system. In: 2016 18th Asia-Pacific network operations and management symposium (APNOMS). IEEE, , pp 1–4
19. Giang NK, Blackstock M, Lea R, Leung VC (2015) Developing IoT applications in the fog: a distributed dataflow approach. In: 2015 5th International conference on the internet of things (IOT). IEEE, pp 155–162
20. Yousefpour A, Patil A, Ishigaki G, Kim I, Wang X, Cankaya HC, Zhang Q, Xie W, Jue JP (2019) FogPlan: a lightweight QoS-aware dynamic fog service provisioning framework. IEEE Internet Things J 6(3):5080–5096
21. Aazam M, Huh E-N (2015) Dynamic resource provisioning through fog micro datacentre. In: 2015 IEEE international conference on pervasive computing and communication workshops (PerCom workshops). IEEE, pp 105–110
22. Aazam M, Huh E-N (2015) Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT. In: 2015 IEEE 29th international conference on advanced information networking and applications. IEEE, pp 687–694
23. Alsaffar AA, Pham HP, Hong C-S, Huh E-N, Aazam M (2016) An architecture of IoT service delegation and resource allocation based on collaboration between fog and cloud computing. Mob Inf Syst 2016
24. Tao Y, Wang X, Xu X, Chen Y (2017) Dynamic resource allocation algorithm for container-based service computing. In: 2017 IEEE 13th International symposium on autonomous decentralized system (ISADS). IEEE, pp 61–67
25. Ni L, Zhang J, Jiang C, Yan C, Yu K (2017) Resource allocation strategy in fog computing based on priced timed petri nets. IEEE Internet Things J 4(5):1216–1228
26. Hoang D, Dang TD (2017) FBRC: optimization of task scheduling in fog-based region and cloud. In: 2017 IEEE Trustcom/BigDataSE/ICESS. IEEE, pp 1109–1114
27. Zhang H, Xiao Y, Bu S, Niyato D, Yu FR, Han Z (2017) Computing resource allocation in three-tier IoT fog networks: a joint optimization approach combining Stackelberg game and matching. IEEE Internet Things J 4(5):1204–1215
28. Desikan KS, Srinivasan M, Murthy CSR (2017) A novel distributed latency-aware data processing in fog computing-enabled IoT networks. In: Proceedings of the ACM workshop on distributed information processing in wireless networks, pp 1–6
29. Skarlat O, Nardelli M, Schulte S, Dustdar S (2017) Towards QoS-aware fog service placement. In: 2017 IEEE 1st international conference on fog and edge computing (ICFEC). IEEE, pp 89–96
30. Mahmud R, Srirama SN, Ramamohanarao K, Buyya R (2019) Quality of Experience (QoE)-aware placement of applications in fog computing environments. J Parallel Distrib Comput 132:190–203
31. Brogi A, Forti S, Guerrero C, Lera I (2019) How to place your apps in the fog-state of the art and open challenges. arXiv preprint arXiv:1901.05717
32. Abbasi M, Pasand EM, Khosravi MR (2020) Workload allocation in IoT-fog-cloud architecture using a multi-objective genetic algorithm. J Grid Comput 18:43–56. https://doi.org/10.1007/s10723-020-09507-1
33. Gupta H, Dastjerdi AV, Ghosh SK, Buyya R (2017) iFogSim: a toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments. Software Pract Experience 47(9):1275–1296
34. Zeng D, Gu L, Guo S, Cheng Z, Yu S (2016) Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system. IEEE Trans Comput 65(12):3702–3712

35. Lin BY, Shen H (2015) Cloud fog: towards high quality of experience in cloud gaming. In: 2015 44th International conference on parallel processing. IEEE, pp 500–509
36. Wang N, Varghese B, Matthaiou M, Nikolopoulos DS (2017) ENORM: a framework for edge node resource management. IEEE Trans Serv Comput

Improvement of Task Offloading for Latency Sensitive Tasks in Fog Environment

Parmeet Kaur and Shikha Mehta

Abstract Fog computing is gaining rapid acceptance as a distributed computing paradigm that brings cloud-like services near the end devices. It enhances the computation capabilities of mobile nodes and IoT (Internet of Things) devices by providing compute and storage capabilities similar to the cloud but at a lower latency and using lesser bandwidth. Additional advantages of fog computing include its support for node mobility, context awareness, reliability and scalability. Due to its multiple benefits, fog computing is used for offloading tasks from applications executing on end devices. This allows faster execution of applications using the capabilities of fog nodes. However, the task offloading problem in the fog environment is challenging due to the dynamic nature of fog environment and multiple QoS (Quality of Service) parameters dependent on the application being executed. Therefore, the chapter proposes a QoS-aware task offloading strategy using a novel nature-inspired optimization algorithm, known as the Smart Flower Optimization Algorithm (SFOA). The proposed strategy takes into account the QoS parameters such as the task deadlines and budget constraints in selection of appropriate fog nodes where computation tasks can be offloaded. The proposed strategy has been simulated and the results have verified the efficacy of the strategy.

Keywords Task offloading · Fog computing · IoT · Smart Flower Optimization Algorithm · Particle Swarm Optimization · Shuffled Frog Leaping Algorithm

P. Kaur (✉) · S. Mehta
Department of Computer Science & Enginnering and Information Technology,
Jaypee Institute of Information Technology, Noida, India
e-mail: parmeet.kaur@jiit.ac.in

S. Mehta
e-mail: shikha.mehta@jiit.ac.in

R. Tiwari et al. (eds.), *Energy Conservation Solutions for Fog-Edge Computing Paradigms*,
Lecture Notes on Data Engineering and Communications Technologies 74,
https://doi.org/10.1007/978-981-16-3448-2_3

1 Introduction

Mobile nodes and sensor devices are finding a number of applications in today's smart world. Apart from collecting rich data, many sophisticated tasks are also being executed on such devices. These tasks or applications are usually data-driven and require computational power to process this data. Since mobile devices and sensors may face resource constraints in terms of limited storage and compute powers, cloud computing has been used to support these devices. Tasks requiring computation power or data exceeding the storage available at mobile devices or sensors are transferred to cloud servers for successful execution. The transfer of data and related tasks from a resource-constrained device to external servers is referred to as offloading [1, 2].

Cloud computing provides multiple benefits in context of task offloading in resource-constrained environments. The most important of these benefits is the availability of a vast pool of heterogeneous resources that can be leveraged for computation of the required tasks [3]. The obtained results are the transferred to the destination nodes. However, use of cloud computing is not preferable for latency-sensitive tasks. This is due to the centralized nature of cloud computing as a result of which transfer of tasks and subsequently the results involves latency. The amount of latency depends on the bandwidth availability for communication between the nodes (mobile or sensor) and the cloud servers. This delay makes cloud-based processing infeasible for many tasks. Further, larger size tasks may take a longer time in transferring the cloud. Therefore, cloud may not be a viable solution for tasks that need immediate execution and cannot afford latency. Instead, there is a need for computing resource availability closer to the nodes so that results can be obtained in desired time [4].

Fog computing is emerging as a possible solution for the offloading problem for latency-sensitive tasks. It is a distributed computing paradigm that acts as an intermediate layer between offshore Cloud data centres and IoT devices or sensors/mobile nodes [5]. Fog devices are placed at the edge of network and closer to the IoT devices as compared to cloud servers. Fog computing provides computation, networking and storage facilities to resource-constrained devices so that cloud-like services can be extended closer to the IoT devices, sensors or mobile nodes. The concept of Fog computing was first introduced by Cisco in 2012 to address the challenges of IoT applications in conventional Cloud computing. It is related to the concept of Edge computing. If fog devices are used for processing the tasks, it will reduce the latency in processing significantly. Further, this also helps in alleviating the network congestion that can occur when a large number of IoT devices transfer data and tasks simultaneously to the cloud servers. In comparison, a judicious selection of fog nodes for task offloading by specific devices can prevent network congestion.

This chapter presents a Smart Flower Optimization Algorithm (SFOA) [6] based task offloading approach for latency-sensitive tasks in fog environment. Tasks that need quick processing are offloaded to selected Fog nodes in the system. The selection of fog nodes for specific tasks is done by using SFOA which is a new and robust nature-inspired algorithm for solving optimization problems. Nature-inspired algorithms have been found to provide near-optimal solutions in multiple complex problems such as resource allocation, scheduling of workflows, search optimizations, etc. Implementation of SFOA in task offloading problem was postulated to give effective results and this was validated by the simulation results. Its performance has been compared with that of two other well-known nature-inspired algorithms, namely, Particle Swarm Optimization (PSO) and Shuffled Frog Leaping Algorithm (SFLA).

The chapter makes the following contributions:

- Presents task offloading problem as an optimization problem in the IoT-fog environment
- Presents a discrete version of a new nature-inspired algorithm (NIA), the Smart Flower Optimization Algorithm and maps the task offloading problem to it.
- Evaluates the efficacy of SFOA under varying workloads for the considered problem and compares with two other well-known nature-inspired algorithms
- Determines which NIA performs well for a given workload.

The chapter is organized as follows: Sect. 2 provides an overview of the paradigms of cloud, fog and edge computing. Literature review is presented in Sect. 3. Section 4 discusses the Smart Flower Optimization algorithm. The use of WOA for task offloading problem has been described in Sect. 5. Results of simulation experiments are put forth in Sect. 6. Concluding remarks are listed in Sect. 7.

2 Cloud, Edge and Fog Computing

With the growth in technology, computing has become possible on a variety of devices, such as mobile nodes and Internet of Things (IoT) devices. Of these, IoT connects multiple, ranging in hundreds to billions, of heterogeneous nodes through a network for gathering data from the underlying system. This data is analyzed and processed to gain insights about the system. IoT has led to development of applications such as smart transportation, smart cities, smart homes, smart agriculture, waste management, etc. [7, 8]. Similarly, many applications such as gaming, video processing, etc. are being commonly executed on mobile devices. However, despite the advancements in technology, IoT and mobile devices are still resource-constrained. These devices possess limited storage, computing and networking capabilities and may not be able to independently process vast amounts of data or execute compute-intensive tasks within a given time frame. If size of data or complexity of applications increases, these devices are unable to execute these applications within a given time.

Cloud computing has been used to perform computing and storage tasks for these devices. Cloud-based servers can support data storage and computations by providing resources on a pay-per-use over the Internet. However, computing on cloud, it being a centralized infrastructure, causes a delay in execution. Applications such as sprinkling water in response to fire detection, changing settings of devices, opening a valve if pressure reading exceeds a threshold, sending an alert to a health worker in case an anomaly is detected, etc. cannot involve such delays in execution. Further, the growth in number of IoT devices also increases the chances of network congestion as the size of data being transferred increases. Therefore, there is a need to minimize the processing time for latency-sensitive applications while also utilizing the computing resources efficiently.

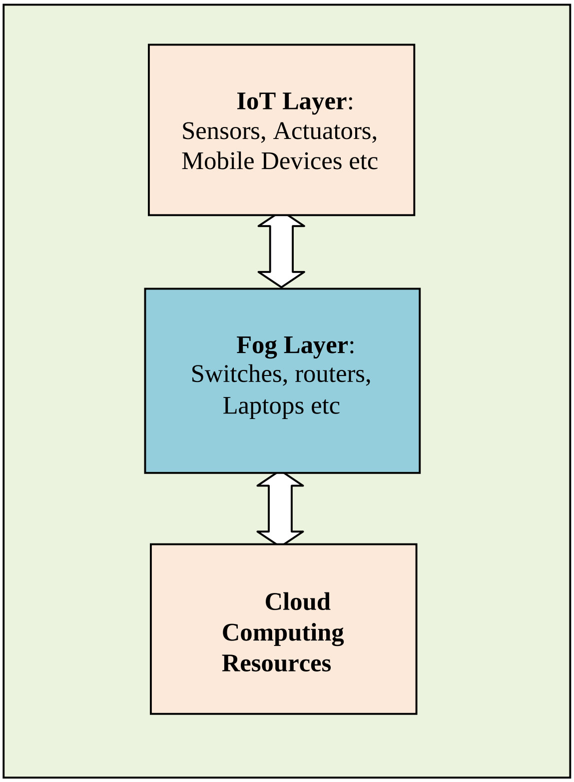
One method that has emerged to timely process latency-sensitive applications is the use of special nodes, known as fog nodes placed in the local network near the IoT devices or the mobile nodes. Tasks can be offloaded to a suitable fog node that has the adequate processing and storage resources. Though a fog node cannot match the resource availability of a cloud, yet a number of fog nodes collectively can be employed to meet the computing requirements of a given set of IoT devices/mobile nodes. These fog components are equipped with higher computing, storage or networking capabilities and are able to support execution of latency-sensitive tasks not possible on the IoT devices themselves. Network switches, routers, Raspberry Pi, etc. can act as fog nodes. Fog computing is related closely to the concept of edge computing. An edge network comprises of end devices, such as mobile phones, smart devices, etc. or the edge devices such as set-top boxes, bridges, base stations, wireless access points, etc. and the edge servers [9]. Edge computing refers to computing at the device itself, while fog computing causes computing to take place in the local network of the device. The use of fog nodes is beneficial, especially, for latency-sensitive applications as the round-trip time to cloud is eliminated. Cloud can be used to store summaries and statistics periodically from multiple fog nodes and also for executing applications that can take longer times to execute.

A typical fog computing system is illustrated in Fig. 1 and used in the current work. It is a three-layered architecture, where the IoT devices such as the sensors, actuators or the mobile devices form the first layer. The fog nodes comprise of the second or intermediate layer which takes input (in the form of tasks or data) from the IoT devices and subsequently returns results or values to them. Cloud computing resources are the third layer in the architecture. Each layer is connected to the other through some communication medium.

Fog computing nodes are located closer to the IoT, they act as an intermediate layer between the IoT nodes and cloud computing. We assume a flat organization of fog nodes in the current work. However, they can form dynamic or static Clusters based on their location or the application requirements. Similarly, master-slave organization may also be used where a master Fog node can coordinate the functions, workload distribution, data flow, etc. of the slave nodes. In the current problem, the IoT devices should make a judicious selection of fog nodes for task offloading. This involves checking the availability of required resources along with the latency incurred in the offloading process.

There can be several QoS parameters that need to be met while offloading tasks. Sine latency-sensitive tasks are being considered here, latency of response from the fog node is the most important parameter. It is imperative that the results are delivered from the fog nodes to IoT devices within an acceptable time interval that is defined as the maximum tolerable latency of a task. Another QoS parameter relates to the

Fig. 1 System model



cost involved in use of fog nodes. The cost is incurred when fog nodes are deployed in the network as well as during their operation. Hence, fog nodes can be used only if the gains exceed these costs. Similarly, there can be QoS parameters related to user requirements of security, privacy, etc.

Since the selection of fog nodes for task offloading is guided by multiple objectives along with the QoS parameters, it is a complex and challenging problem. Nature-inspired algorithms have often been used for solving such problems. Hence, this work explores the Smart Flower optimization algorithm for finding a near-optimal solution to the task offloading problem. Selection of a suitable computing device for each delay-sensitive task is a challenge addressed by the proposed solution using Smart Flower optimization algorithm (SFO) technique. The proposed SFO based strategy finds an optimal computing device (i.e., fog device) for each real-time task using multiple Quality-of-Service (QoS) parameters, namely cost and resource utilization.

3 Literature Review

The approach to offload tasks from resource-constrained devices to fog devices has garnered significant research attention in last few years [5, 10]. Different works have focussed on optimization of various goals with respect to offloading. A method to offload IoT applications with an objective to minimize the execution time along with the energy consumption of the resources has been proposed in [11]. The offloading strategy in [12] focussed on reducing the latency of applications and energy usage of the fog devices. A few other offloading approaches, such as [13, 14] have also been developed for minimizing the energy consumption of resources and the execution time of the applications. An offloading approach specific to latency-sensitive applications has been presented in [15] with the aim of decreasing the time required for computation by the applications. The framework proposed in [16] endeavours to minimize the service delay of IoT applications executing in fog-cloud environment support. Based on the size of a task, the framework ensures that the delay that will be caused by its execution on a fog node is less than a threshold of time. Otherwise, it finds another fog node that will complete the execution of the task within the stipulated time. If no such fog node is found, the task is transferred to a cloud server. Another offloading approach for delay-sensitive applications has been proposed in [17].

Workload from IoT nodes is divided among fog nodes and cloud servers by the work in [18] in a way that reduces the consumption of power and the delay incurred in service delivery. In [18], an analysis of the workload allocation between the fog nodes and the cloud nodes is investigated to minimize the power consumption under service delay constraints. This approach assumes connection and cooperation between fog nodes and cloud servers.

The authors of [19] considered balancing the load distribution among fog nodes while offloading tasks to these nodes. This is expected to give an efficient application execution. In comparison, the work in [20] is based on tasks' priority in a system that assumes the presence of fog nodes as well as the cloud services. By prioritizing the tasks, waiting time and delay of tasks that need to meet a deadline are reduced.

Due to the complexity of this problem, several heuristic and meta-heuristic-based solutions have been presented in literature. The work in [21] presents a task offloading strategy for IoT applications using task classification and calculating cost of execution. However, the authors have not accounted for the cost incurred in communication amongst tasks. In comparison, a genetic algorithm-based single objective data offloading approach has been proposed in [22]. The objective is to minimize the overall cost of data offloading in a fog-based system by reducing the costs of execution as well as communication between tasks. In comparison, the work in [23] aims to minimize the overall time and cost of application execution. A few other approaches of task offloading [24, 25] have also paid attention to efficient resource utilization in the fog environment. The work in [26] utilizes Bee Swarm algorithm for optimizing the execution time along with efficient resource usage. Similar objectives are achieved using a genetic algorithm (GA) based offloading approach in [27].

Further, the approach of Particle Swarm Optimization (PSO) is utilized in designing an offloading strategy that optimizes the energy consumption in the fog environment along with the latency in [28]. Authors of [29] have presented a deadline-aware offloading approach that uses both GA and PSO to optimize the delay and execution time of applications.

Another swarm optimization algorithm, Ant Colony Optimization (ACO) is employed in [30] for scheduling tasks of IoT applications on the fog nodes. The proposed approach aims to reduce the delay of task execution using profiling the IoT tasks. A strategy to optimize operational time and cost using a genetic algorithm is presented in [31]. Another genetic algorithm based offloading strategy in fog systems is put forth in [32]. The work in [33] employs a variation of the PSO algorithm to select an appropriate fog for each real-time IoT task. The work considers the parameters of cost and resource utilization. An Ant Colony Optimization based task distribution over the fog nodes is proposed in [34] to enhance quality of service and reduce the task response time. The present work makes use of a recent nature-inspired algorithm for the considered problem.

4 Smart Flower Optimization Algorithm

The Smart Flower Optimization Algorithm (SFOA) is a meta-heuristic technique inspired by the growth behaviour of immature sunflowers. In this algorithm, the stem length of the sunflower represents the individual solution of the population. The growth of stem length in sunflowers is determined by the internal mechanisms like direction of the sun in the daytime and biological clock all through the night. It is well known that direction of the immature sunflowers changes according to the sun such that during the day, immature sunflowers align themselves in the east and gradually turn towards the west with the change in direction of the sun. After the sun set, they swing themselves again towards the east waiting for the sunrise. These heliotropic movements are only shown by the immature sunflowers. For simulation, the growth of immature sunflower is modelled mathematically along two dimensions: Sunny day vs cloudy or rainy day. As the growth pattern of immature sunflower is different on cloudy or rainy days as compared to sunny days.

In SFOA, stem length of the immature sunflower represents the potential solution to the problem to be solved. Since SFOA is used to solve optimization problems, objective of the problem represents the fitness function that indicates the stem length of the sunflower. Longer the stem length better is the solution. The evolved stems length or novel individuals are created according to the internal mechanism that is heliotropic movements, which are responsible for complete growth of the sunflower during the day in the decision space decision space. The detailed steps of SFOA are shown in Fig. 2. As SFOA is swarm intelligent technique which begins with generation of random population of initial potential solution depending upon the problem to be solved. In the next step fitness of all potential solutions is computed and best

Input: population size(P), maximum number of iterations(Itr_{max}), number of dimensions(dim), sun parameter(sun)

Output: Best solution(L_{best}), Fitness of the best solution(f_{best})

//Main Algorithm
Generate initial population of size P randomly.

Identify and Store fitness of best solution or best length as $L_{best.}$
For Itr=1 to Itr_{max}
$d = damping_{max} - Itr * \left(\frac{(damping_{max} - damping_{min})}{Itr_{max}}\right)$ **//Generate the damping factor(d)**
For i=1 to P
Generate parameter ω where ω is the angle of sine function $\in [0, 160^{o}]$
For j =1 to dim
If sun= =1 //sunny day
Aux $\in [0, 1]$ //growth harmone
Hrs = $\in [0, 100]$ // biological clock
If Hrs<=24
$L_{new\,SF}^{Itr+1} = L_{old\,SF}^{Itr+1} + d * sin(\omega) * [Aux * L_{best\,SF}^{Itr+1} - L_{old\,SF}^{Itr+1}]$ **(1) //sunny day**
Else
$L_{new\,SF}^{Itr+1} = L_{old\,SF}^{Itr+1} + d * sin(\omega) * [L_{best\,SF}^{Itr+1} - L_{old\,SF}^{Itr+1}]$ **(2) // cloudy or rainy day**
$\varphi \in [0, 0.01]$
$\omega^{j+1} = \omega^{j} + \varphi$
end j
end i
Compute fitness of newly generated population and find $L_{best\ new.}$
If $L_{best\ new} < L_{best.}$
$L_{best\ new} = L_{best.}$
End Itr

Fig. 2 Description of Smart Flower Optimization Algorithm

solution of the population is identified as Lbest. The algorithm is executed for predefined number of iterations. In every iteration, evolution of the candidate solutions is performed according to the growth mechanism of the immature sunflowers.

For simulating the heliotropic movements, parameter “Sun” is used for switching among the sunny and cloudy modes. Value of “Sun” parameter is assigned ‘1’ for sunny day and it is assigned value of ‘0’ for the cloudy/rainy day’s mode. The default mode is Sunny day. In nature, heliotropic movements of the immature sunflower decrease gradually with the change in direction of the sun from sunrise to sunset and stop completely at the mature stage. This behaviour is simulated with the help of a damping factor ‘d’ and is computed using Eq. 1 as shown in the algorithm. Heliotropic movements occurring due to ‘Auxin’ hormone are represented using the sine function. ‘Auxin’ is a growth hormone responsible for development of sunflower and stem elongation along with inducing the heliotropic movements during a 24 h day and night cycle. In the SFOA, ‘Aux’ variable represents ‘Auxin’ and its value is chosen in the range of [0, 1]. Since heliotropic movements happen only during the day, updates in the stem length are represented using Eq. 1 whenever hours of the day are less than 24. Equation 1 incorporates the effects of Auxin hormone using

'Aux' parameter. The angles made by the heliotropic movements are defined using parameter 'ω'. To depict the random movements of sunflower in odd hours, value of 'ω' is updated using phase angle 'φ'. The value of 'φ' is decided at random in the range [0, 0.01]. After evolution, fitness of new stem length is compared with the current best, if it's better, it is updated else old fitness value is retained. This whole process is carried out for predefined number of iterations. In the presented work SFOA is adapted to solve task offloading problem in fog environment. For the same, fitness or length of the flower is computed using Eq. 2.

5 Application of SFOA to Task Offloading Problem

The present work has utilized SFOA finding an optimal solution to the task offloading problem in the fog environment. SFOA is a meta-heuristic-based algorithm in which population comprises of immature sunflowers each of which is a n-dimensional search agent. The algorithm needs to be mapped to the task offloading problem for implementation.

In SFOA, every immature sunflower in the population is assumed to possess a stem length which represents the Dim-dimensional search space. We have developed a discrete version of the basic SFOA to obtain a solution to the considered problem. Therefore, we have used n-dimensional sunflowers in the population where n indicates the number of tasks for each workload that need to be offloaded to m fog devices. The value of each dimension of the immature sunflower is an integer between 1 and m, denoting the fog device on which the corresponding task is offloaded.

Consider the example of Fig. 3, if 9 tasks are required to be offloaded on 3 fog nodes, then each search agent, i.e., an immature sunflower, will be 9-dimensional where each dimension takes a value of 1, 2, or 3. Now, if the value of third dimension in a search agent is 1, it shows that the third task in the workload is offloaded to the fog node, 1 in a given solution. The fitness of each solution is computed in SFOA by using the fitness function described next.

Every nature-inspired algorithm possesses a fitness or objective function that is used to evaluate the goodness of the solution. In SFOA, the stem length of every immature sunflower denotes a potential solution to the optimization problem. Every flower is associated with the objective function to compute its fitness. A sunflower with a longer stem implies better fitness value and therefore, a better solution. Since presented work aims to reduce the execution cost of a task workload offloaded to the fog devices, the considered problem is formulated as a minimization problem. The

Structure of Search Agent									
Task No	1	2	3	4	5	6	7	8	9
Fog Device ID	2	3	2	1	3	2	1	2	1

Fig. 3 Mapping of task to fog devices

fitness value of a sunflower, F_{sf} is evaluated by computing the total cost of workload execution. This is, in turn, the sum of execution cost of each task in the workload on the corresponding fog node in the solution. The aim is to select those fog nodes for executing tasks which result in the minimum execution cost. Therefore,

$$F_{sf} = \sum_{i=1}^{n} \mathrm{TEC}_i \tag{1}$$

where TEC_i represents the task execution cost of task, i on the fog node as indicated by the sunflower. Further, we have also computed the total execution time (TET) of each solution for compliance with the QoS parameter of execution deadline. Any solution, regardless of its execution cost is discarded if the TET exceeds the provided deadline.

$$\mathrm{TET} = \mathrm{FT}_i \tag{2}$$

where FT_i denotes the finish time of a task, i in the workload. Thus, the designed fitness function will provide only those solutions that adhere to the deadline, as specified for the application.

6 Simulations and Results

The efficacy of the proposed task offloading strategy using SFOA is evaluated by simulation experiments. The simulation experiments have considered different workload-Fog scenarios for evaluation. The performance of proposed approach has been studied for both small as well as large size workloads. Depending on the size of workloads, number of fog devices has been varied too. The performance of SFOA has been compared with two well-established nature-inspired algorithms, Particle Swarm Optimization (PSO) algorithm and Shuffled Frog Leaping (SFLA) algorithm [35]. All the algorithms have been executed on a Java simulator developed by the authors. Considering the random or non-deterministic nature of the nature-inspired algorithms, each simulation was performed 25 times and the results reported in the chapter are an average of the obtained results.

Particle swarm optimization (PSO) [36] is a well-established population-based nature-inspired algorithm used as a stochastic optimization technique. The algorithm was designed taking an inspiration from the flocking behaviour of birds and a similar schooling nature of fish in search of food. PSO is an evolutionary algorithm like the Genetic Algorithms (GA). The initial population of the system represents a set of random solutions and this is evolved iteratively in search of the optima. Each potential solution is called a particle and moves through the problem space by following the particles currently nearest to the optimal solution (or the food). In this manner, all particles eventually converge to the optimal solution. PSO is an easy-to-implement algorithm that has been frequently used in several optimization problems in diverse areas, such as resource scheduling, clustering, control system design, search engines, etc.

Similar to PSO, the Shuffled Frog Leaping Algorithm (SFLA) [37] is also a nature-inspired algorithm based on the behaviour of frogs searching for food. SFLA is based on Metaheuristic memetics and has also been employed for solving multiple optimization problems in literature. It is a population-based algorithm that utilizes a local search space exploration method within a memeplex and a global exploration using communication between frogs belonging to different memeplexes. The initial population of frogs is set randomly and subsequently sorted in descending order based on their respective fitness. Frogs are then divided into memeplexes. The population evolves iteratively in SFLA by directing the worst solutions towards best local/global solutions.

The simulations were performed to evaluate the execution cost when tasks are offloaded to fog nodes using the considered algorithms. In order to assess the scalability of algorithms, the size of workloads was varied. The first experiment evaluated the SFOA for workloads with a large number of tasks. The number of tasks in the workload has been varied from 250 to 1000. Execution cost for each workload was computed of the three algorithms. It can be observed from Fig. 4 that SFOA resulted in least execution cost for each workload and outperformed both PSO and SFLA. Percentage improvement in performance for each algorithm is depicted in Table 1.

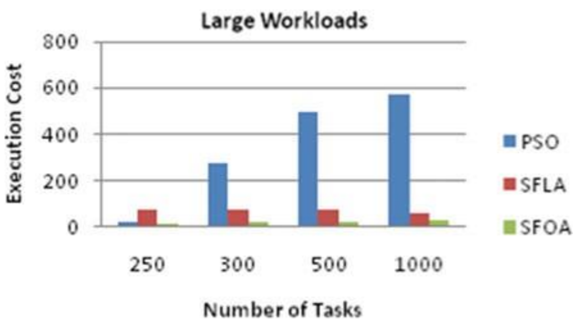


Fig. 4 Comparison of SFOA, PSO and SFLA for large workloads

Table 1 Percentage improvement due to SFOA with respect to PSO and SFLA for large workloads

	Percentage improvement of SFOA with respect to	
Number of tasks	PSO	SFLA
250	7.8	75.8
300	92.9	75.1
500	94.9	68.3
1000	95	55.6

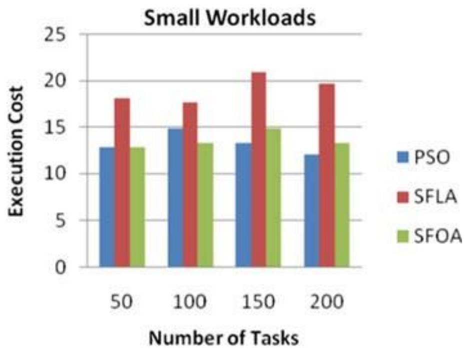


Fig. 5 Comparison of SFOA, PSO and SFLA for small workloads

Table 2 Percentage improvement due to SFOA with respect to PSO and SFLA for small workloads

	Percentage improvement of SFOA with respect	
Tasks	PSO	SFLA
50	0	28.9
100	10.8	25
150	−12.1	28.8
200	−10	32.7

The second experiment computed the execution costs for small-sized workloads using SFOA, PSO and SFLA. Similar to large-sized workloads, SFOA was observed to result in smaller execution cost than SFLA (Fig. 5). However, PSO and SFOA depicted equivalent performance as both performed comparably. Table 2 depicts the percentage improvement due to SFOA as compared to both PSO and SFLA.

From the experiments, it can be inferred that as the size of workload increases, the performance of SFOA based task offloading algorithm improves and it outperforms other algorithms.

7 Conclusion

The chapter has presented the use of a nature-inspired algorithm, namely the Smart Flower Optimization Algorithm (SFOA) for finding a solution to the task offloading problem in IoT-Fog environment. The focus of the work is upon offloading of delay-sensitive tasks that need to be executed in the IoT. Executing them on centralized and off-shore cloud computing servers is not feasible due to the latency incurred. The solution presented in this work finds a mapping between tasks and available fog nodes. The obtained mapping is based on the criterion of reducing the execution cost as well as execution cost such that the provided deadlines for task execution are not exceeded. The performance of the presented algorithm is compared to that of two other nature-inspired algorithms, Particle Swarm Optimization (PSO) and Shuffled Frog Leaping Algorithm (SFLA). Simulation results show that performance of SFOA is comparable to PSO and better than SFLA for small-sized workloads. Moreover, SFOA outperforms SFLA and PSO both for large workloads.

References

1. Zhu Q, Si B, Yang F, Ma Y (2017) Task offloading decision in fog computing system. China Commun 14(11):59–68
2. Sun H, Huiqun Yu, Fan G, Chen L (2020) Energy and time efficient task offloading and resource allocation on the generic IoT-fog-cloud architecture. Peer-to-Peer Networking Appl 13(2):548–563
3. Xu X, Liu Q, Luo Y, Peng K, Zhang X, Meng S, Qi L (2019) A computation offloading method over big data for IoT-enabled cloud-edge computing. Futur Gener Comput Syst 95:522–533
4. Aazam M, Islam SU, Lone ST, Abbas A (2020) Cloud of things (CoT): cloud-Fog-IoT task offloading for sustainable internet of things. IEEE Trans Sustain Comput
5. Mukherjee M, Shu L, Wang Di (2018) Survey of fog computing: fundamental, network applications, and research challenges. IEEE Commun Surv Tutorials 20(3):1826–1857
6. Sattar D, Salim R (2020) A smart metaheuristic algorithm for solving engineering problems. Eng Comput, 1–29
7. Stoyanova M, Nikoloudakis Y, Panagiotakis S, Pallis E, Markakis EK (2020) A survey on the internet of things (IoT) forensics: challenges, approaches, and open issues. IEEE Commun Surv Tutorials 22(2):1191–1221
8. Puri V, Kaur P, Sachdeva S (2020) Data anonymization for privacy protection in fog-enhanced smart homes. In: 2020 6th international conference on signal processing and communication (ICSC). IEEE, pp 201–205
9. Shi W, Dustdar S (2016) The promise of edge computing. Computer 49(5):78–81
10. Aazam M, Zeadally S, Harras KA (2018) Offloading in fog computing for iot: Review, enabling technologies, and research opportunities. Futur Gener Comput Syst 87:278–289
11. Wang Y, Wang K, Huang H, Miyazaki T, Guo S (2018) Traffic and computation co-offloading with reinforcement learning in fog computing for industrial applications. IEEE Trans Industr Inf 15(2):976–986
12. Liang K, Zhao L, Chu X, Chen H-H (2017) An integrated architecture for software defined and virtualized radio access networks with fog computing. IEEE Network 31(1):80–87
13. Zhao X, Zhao L, Liang K (2016) An energy consumption oriented offloading algorithm for fog computing. In: International conference on heterogeneous networking for quality, reliability, security and robustness. Springer, pp 293–301
14. Zhang K, Mao Y, Leng S, Zhao Q, Li L, Peng X, Pan L, Maharjan S, Zhang Y (2016) Energy-efficient offloading for mobile edge computing in 5g heterogeneous networks. IEEE access 4:5896–5907
15. Hasan R, Hossain M, Khan R (2018) Aura: an incentive-driven adhoc iot cloud framework for proximal mobile computation offloading. Futur Gener Comput Syst 86:821–835
16. Wang Y, Sheng M, Wang X, Wang L, Li J (2016) Mobile-edge computing: partial computation offloading using dynamic voltage scaling. IEEE Trans. on Commun 64(10):4268–4282
17. Tan H, Han Z, Li X-Y, Lau FC (2017) Online job dispatching and scheduling in edge-clouds. In: INFOCOM 2017-IEEE conference on computer communications. IEEE, pp 1–9
18. Yousefpour A, Ishigaki G, Jue JP (2017) Fog computing: towards minimizing delay in the internet of things. In: Proceedings of IEEE international conference edge computing (EDGE). pp 17–24

19. He J, Wei J, Chen K, Tang Z, Zhou Y, Zhang Y (2018) Multitier fog computing with large-scale iot data analytics for smart cities. IEEE Internet Things J 5(2):677–686
20. Deng R, Lu R, Lai C, Luan TH, Liang H (2016) Optimal workload allocation in fog-cloud computing towards balanced delay and power consumption. IEEE Internet Things J 3(6):1171–1181
21. Fricker C, Guillemin F, Robert P, Thompson G (2016) Analysis of an offloading scheme for data centers in the framework of fog computing. ACM Trans Model Perform Eval Comput Syst (TOMPECS) 1(4):16
22. Adhikari M, Mukherjee M, Srirama SN (2019) DPTO: a deadline and priority-aware task offloading in fog computing framework leveraging multi-level feedback queueing. IEEE Internet of Things J, 1–1
23. Chen Y-C, Chang Y-C, Chen C-H, Lin Y-S, Chen J-L, Chang Y-Y (May 2017) Cloud-fog computing for information-centric Internetof-Things applications. In: Proceedings of international conference applied system innovation (ICASI). pp 637–640
24. Brogi A, Forti S, Guerrero C, Lera I (2019) Meet genetic algorithms in monte carlo: optimised placement of multi-service applications in the fog. In: International Conference on Edge Computing (EDGE). IEEE, pp 13–17
25. Jian C, Li M, Kuang X (2018) Edge cloud computing service composition based on modified bird swarm optimization in the internet of things. Cluster Comput 1–9
26. Bitam S, Zeadally S, Mellouk A (2018) Fog computing job scheduling optimization based on bees swarm. Enterp Inf Syst 12(4):373–397
27. Aryal RG, Altmann J (2018) Dynamic application deployment in federations of clouds and edge resources using a multiobjective optimization ai algorithm. In: 2018 third international conference on fog and mobile edge computing (FMEC). IEEE, pp 147–154
28. Wan J, Chen B, Wang S, Xia M, Li D, Liu C (2018) Fog computing for energy-aware load balancing and scheduling in smart factory. IEEE Trans. on Industrial Informatics 14(10):4548–4556
29. Ezhilarasie R, Reddy MS, Umamakeswari A (2019) A new hybrid adaptive ga-pso computation offloading algorithm for iot and cps context application. J Intell Fuzzy Syst (Preprint), pp 1–9
30. Naqvi SAA, Javaid N, Butt H, Kamal MB, Hamza A, Kashif M (2019) Metaheuristic optimization technique for load balancing in cloud-fog environment integrated with smart grid. In: Barolli L, Kryvinska N, Enokido T, Takizawa M (eds) Advances in network-based information systems. Springer, Cham, pp 700–711
31. Binh HTT, Anh TT, Son DB, Duc PA, Nguyen BM (2018) An evolutionary algorithm for solving task scheduling problem in cloud-fog computing environment. In Proceedings of 9th International Symposium on Infermation and Communication Technology (SoICT). New York, pp 397–404. https://doi.org/10.1145/3287921.3287984
32. Canali C, Lancellotti R (2019) GASP: genetic algorithms for service placement in fog computing systems, algorithms 12(10):201. [Online]. Available: https://www.mdpi.com/1999-4893/12/10/201
33. Adhikari M, Srirama SN, Amgoth T (2019) Application offloading strategy for hierarchical fog environment through swarm optimization. IEEE Internet Things J 7(5):4317–4328
34. Hussein MK, Mousa MH (2020) Efficient task offloading for iot-based applications in fog computing using ant colony optimization. IEEE Access 8:37191–37201
35. Dai Lu, Wang B, Yang LT, Deng X, Yi L (2019) A nature-inspired node deployment strategy for connected confident information coverage in industrial Internet of Things. IEEE Internet Things J 6(6):9217–9225
36. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks, vol 4. IEEE, pp. 1942–1948
37. Eusuff M, Lansey K, Pasha F (2006) Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. Eng Optim 38(2):129–154

38. Sharif MU, Javaid N, Ali MJ, Gilani WA, Sadam A, Ashraf MH (2018) Optimized resource allocation in fog-cloud environment using insert select. In: International conference on network-based information systems. Springer, pp 611–623
39. Kaur P, Mehta S (2017) Resource provisioning and work flow scheduling in clouds using augmented Shuffled Frog Leaping Algorithm. J Parallel Distrib Comput 101:41–50

A Sustainable Energy Efficient IoT-Based Solution for Real-Time Traffic Assistance Using Fog Computing

Bhawna Suri, Shweta Taneja, Sunny Kumar, and Sumit

Abstract The ever-growing number of vehicles brings forth challenges in traffic management. This causes various traffic management issues in urban cities around the world. Some of the issues are: delay in emergency/alarming situations, non-deterministic waiting time of local transport, increased fuel consumption, etc. To help the people travelling by local transport in the cities by knowing the position of the bus, at a specific time, would ease them from indefinite wait or pass over of bus. In this chapter, our focus is to provide a trouble free, smart and innovative IoT-based traffic assistant that can solve real time transport related problems. A **Hierarchical Peer Connected Fog Architecture (HPCFA**) is proposed to lower latency time and computational overhead. In HPCFA, the fog nodes are organized in a hierarchy where the peer fog nodes present at the same level are also interconnected with each other. The data from IoT devices equipped on the roads will capture the position of the vehicle which is then transmitted to the nearest fog node. This fog node will further transmit the information through HPCFA to the user. Using HPCFA, the total energy consumption is also reduced to some extent. The proposed architecture is very flexible, as it works both with fog nodes or without fog nodes and directly with the cloud. Further, an android application is also developed for the proposed architecture. The simulations and results are also displayed.

Keywords Cloud computing · Fog computing · Vehicular fog computing (VFC) · RFID (Radio-frequency identification) · Global positioning system (GPS)

B. Suri · S. Taneja (✉)
Department of Computer Science and Engineering,
Bhagwan Parshuram Institute of Technology, GGSIPU, Delhi, India
e-mail: shwetataneja@bpitindia.com

B. Suri
e-mail: bhawnasuri@bpitindia.com

S. Kumar · Sumit
Department of Electronics and Communication Engineering,
Bhagwan Parshuram Institute of Technology, GGSIPU, Delhi, India

R. Tiwari et al. (eds.), *Energy Conservation Solutions for Fog-Edge Computing Paradigms*,
Lecture Notes on Data Engineering and Communications Technologies 74,
https://doi.org/10.1007/978-981-16-3448-2_4

1 Introduction

In India, automotive industry is a major contributor to the Gross Domestic Product (GDP) of the country. In recent years, the Nations' as well as the engineers' aptitude has shown interest in designing and developing various parts of the vehicles to make them Smart Vehicles, for the benefit of the society. The benefit could be in the terms of safety of the passengers travelling via public transport specifically females after the Nirbhaya case which is a shameful tragedy happened in the heart of India. Besides this, there are other benefits like boarding on the public or private vehicles, vehicle theft, tracking and tracing of the valuable items, etc. Nowadays most of the population is residing in cities and the majority of the population are connected to the Internet with at least two IoT devices per person. The IoT devices like smartphones, desktops, laptops, etc. are linked wirelessly/wired to the network and have ability to transmit and receive data. This data transmission over network can be further used through these intelligent IoT devices, and will contribute for the effective use of many non-internet-enabled physical devices and other objects. Focusing on this direction, these billion devices and systems contribute to the city's infrastructure in many ways. These IoT devices can communicate and interact over the internet for remote monitoring and controlling of many devices and objects, home automation, traffic management, water and gas monitoring health care and many more. Thereby if this huge data can be integrated, managed and analysed, it will provide smart solutions for the clean and sustainable environment for the citizens living in the cities or metropolitan cities [1]. If all these smart solutions are well implemented and executed, then automatically the city would become Smart City, hence Smart States and Smart Country.

Above mentioned objectives and many more, are well known objectives of the Smart Cities Mission in many countries and are also the objectives of the Government of India. In this chapter, our focus is on the public transport system—Delhi Transport Corporation (DTC) buses in Delhi. India's capital, Delhi, spans nearly 1483 km^2 with population of almost 16.8 million people and have translated into high levels of demand for motor vehicles [2]. With the increase in the personal mobility needed by the majority of the females going to their workplace, there is an increase in the demand for car ownership and has now become the social status symbol also. Yet the public road transport, buses are still holding the position of being the most popular means of road transport, that caters to about 60% of Delhi's total demand [3]. Delhi's public bus transport system, Delhi Transport Corporation (DTC), is a major bus service provider which operates the world's largest fleet of CNG-fuelled buses. A scheme of cluster buses is operated by Delhi Integrated Multi-Modal Transit System (DIMTS), which is a joint venture of Government of National Capital Territory (NCT) Delhi and IDFC, that are working on DTC with the aim that majority of the people should travel by public transport in comparison to personalized vehicle. To achieve this aim, DIMTS is working on deriving the mechanism to deliver the safe, reliable, accessible and sustainable public transport which can be directly mapped with the increase in demand. With this aim, DIMTS has installed GPS devices with GPRS/3G/4G connectivity in most of the buses to ensure the real-time tracking of the buses. The GPS enabled Automatic Vehicle Location (AVL) allows real-time tracking of bus movement and provides information on its location and further contributes in calculating the speed of the bus, route of the bus followed, etc.

Among the total DTC buses, still there are many existing buses of DTC in which GPS is not mounted. The installation of GPS in all DTC buses might not be feasible because they were not built with this aim at that point of time and installation of GPS in all the buses would be a costly budget. Among all the above-said transport problems, our aim is that, if a passenger wants to travel from a point P_i–P_j via DTC bus and is standing at bus stop B_k. He should know about the following:

1. The list of buses $\{D_1, D_2, \ldots, D_n\}$ taking him/her from Point P_i–P_j.
2. Standing at a bus stop B_k if he/she wants to travel via bus D_m ε $\{D_1, D_2, \ldots, D_n\}$, then the current location and estimated arrival time (ETA) of D_m must be known to him to ease him for indefinite wait.
3. The safety of the passengers, specifically the females, the feature of handling the emergency situations are also taken into account.

The easiest way to handle the above-stated issues is by using the smartphone of the driver, that communicates with our Android Application which is installed on the user's/passenger's mobile phone via GSM Vehicular Fog Network.

In this chapter, we have proposed a Hybrid Peer Connected Fog Architecture (HPCFA). It is a hierarchical structure, that contains cloud, fog nodes and IoT devices. The cloud is at the root, fog nodes are present at the next level, that are further interconnected with the other fog nodes at the same level.

1.1 Vehicle Tracking System

A vehicle tracking system is a system where the different hardware devices and software collaborate to provide the location of any vehicle. The vehicle's location can be tracked through various IoT devices like cameras, smart mobile phones or other smart devices through GPS satellites. In this era of technology, the tracking system, or locating system, can also be done with automatic vehicle location softwares. The information of vehicles can be seen on digital maps also. These systems are used for surveillance purposes both for smart objects and human beings.

Generally, the basic components of tracking system are—hardware unit or vehicle unit, fixed base station and software unit [4].

- Vehicle Unit or Hardware Unit—The hardware unit is the hardware part namely GPS with GPRS/3G/4G, cellular WIFI modems mounted on the vehicles which are to be tracked or there exists embedded modems complete with SIM card holder, I/O and antenna connectors for transmitting the GPS data.
- Fixed Based Station—It consists of a wireless network system that receives and transfers the information to the data centre. The based station contains software

and geographic map useful for locating the vehicle. The system uses Global Positioning System (GPS), to find information about the position of the vehicle that is to be monitored and then send the latitude and longitude to the monitoring centre through satellite. Maps of every city are available in the based station that has an in-built web server.

- Database and Software—These are used to give the location of the vehicle, that is the coordinates of each visiting point that is saved in the database, which can be later displayed in a screen using Google maps. However, to view the location the vehicle has travelled, the users have to connect themselves to the web server.

These modern software and hardware devices help to locate and trace the vehicles both in online and offline mode.

Hence in today's era, the vehicles are not just the dumb vehicles, rather are smart vehicles equipped with various intelligent sensors for various reasons like theft safety of the vehicle or monitoring on the road, speed calculation and many more. These in-vehicle sensors, communication sensors and on-board units with computation and storage capabilities are termed as Vehicular Fog Computing [5]. In this architecture, there is a better communication between the vehicles and the cloud or fog data centres to get the location awareness and real-time response in conventional vehicular networks. Fog computing is an extension to the concept of cloud computing that shifts the load from the centre to edge networks. There are numerous advantages of fog computing, but there is still a challenge of universal connectivity as well as low latency in real-time traffic management. Vehicular Fog Computing (VFC), a combination of fog computing and vehicular networks, is a promising solution to achieve real-time and dynamic locations of the vehicles.

1.2 Need of Vehicle Tracking

The vehicle tracking systems have helped in turning the traditional transport management system as a Smart Transport Management system by remotely managing the vehicles for the benefit of the society. A good public transport system improves the quality of life of the people by providing better mobility, accessibility, updates on the uncertainty of arrival time of the next bus/metro, etc. The following are the aims/objectives of vehicle tracking:

- The customer as well as the business owner can locate and trace the packet of delivery in their smart phones.
- It can help in efficient parking of vehicles.
- Smallest congestion-free route from source to destination can be identified.
- Monitoring and tracing of vehicles rented for the cab facility.
- Students need to reach on time for their classes, employees can notify the managers or the clients about the delay.

1.3 Problem Statement

This chapter aims to provide real-time information to the passenger travelling via public transport—DTC bus, in the following ways:

1. Assisting the passengers to get the list of buses taking him from his location to the desired destination.
2. Also, if the passenger wants to travel by a specific bus passing through his bus stop, then he will be able to find the Estimated Arrival Time (ETA) of the next incoming bus.

The organisation of this chapter is as follows: Sect. 2 gives the related work. In Sect. 3, concepts like IoT, Cloud computing, Fog computing, etc. are discussed. In Sect. 4, our proposed architecture—Hybrid Peer Connected Fog Architecture (HPCFA) is discussed in detail. Section 5 shows simulations and results with the help of Android application. Section 6 concludes the chapter.

2 Related Work

In the field of vehicle tracking, many authors have given solutions to different problems that have come.

A few of them are presented here.

In [6], authors have designed a localization system for buses, that consists of route management as well as real-time tracking. The system is made for Brazil. It takes data from GPS devices and sends an android application. In another work [7], authors have tried to find delay in buses in Brazil. This is done by comparing the patterns of interleaved events for the departure and arrival of buses. An android based application is designed in [8], that provides current live location of the bus to the user. It also gives the estimated time that bus will take to reach the destination. A similar tracking system is proposed by authors in [9], that gives the live location of the bus to save the waiting time for users. In [10], authors have proposed a system that helps to monitor a school bus, specifies speed of the driver and sends messages if it exceeds the speed limit. It is based on a GPS system along with Raspberry pi.

In the field of multi-vehicle tracking for smart transportation, many authors have contributed their work. In [11], authors have proposed a multi-vehicle detection and tracking in real-time system. It is based on vehicle subfeatures selection and grouping. In [12], authors have proposed a novel tracking method that is based on linked vehicles. Each vehicle maintains a list of positions of other vehicles.

Another related concept that is becoming popular nowadays is that of Vehicular Fog Computing (VFC) in smart cities. It is a blend of fog computing and vehicular networks that provides real-time traffic management. In [13], authors have presented a VFC architecture. The architecture is made up of three layers—fog layer, cloudlet layer and cloud layer. The fog layer consists of vehicles as fog nodes. The next cloudlet layer gets data from vehicles, performs operations and sends to the cloud layer. The cloud layer contains centralized server that manages the overall traffic monitoring system. In [14], authors have proposed an architecture that consists of vehicles grouped in parking, fog node and centralized cloud. They have tried to solve the problem of energy-efficient software matching. In [15], authors have discussed and analyzed four cases of parked as well as moving vehicles in VFC. In another work [16], authors have proposed fog-enhanced vehicular services. It reduces the load on the cloud and also helps in resource estimation and allocation.

Different researchers have given their proposals in the area of traffic management, but to the best of our knowledge, no one has given the real-time location of public transport like DTC bus and Estimated Arrival Time (ETA) of the bus as is needed by the passengers. Also, the list of buses passing through a particular bus stop, that would take a passenger to the desired destination. In this work, we have worked on these aspects of the problem.

3 Background

3.1 IoT

3.1.1 What is IoT?

IoT or Internet of Things is a network of interconnected devices, machines, etc. that can transmit data over the internet. The devices can communicate with each other through mobile app or web browsers. Today IoT is becoming popular technology in almost every domain. It has been applied in various areas like agriculture, smart home, industry, transportation, smart health, vehicle parking and many more [17]. Figure 1 shows applications of IoT devices.

An important application of IoT devices is in the area of smart health. Smart wearable IoT devices are used to improve health care [18]. Ellaji et al. in 2020 have proposed a narrow band IoT technology that can connect multiple intelligent devices together in a health care system [19]. In another work [20], authors have presented a health monitoring system using Fog and IoT for elderly people. The proposed system inputs signals from the daily routine activities of elderly people. This data helps health care workers to monitor their health and behaviour changes.

Another interesting application domain of IoT devices is smart agriculture or precision agriculture. Tremendous amount of work is going on in this area. IoT technology is used in farming methods, managing farms, livestock, etc. Basically, we are moving towards labour free farms with this technology [21]. Another good work is done by Bu and Wang in 2019 in [22]. The authors have used the concept of deep reinforcement learning along with IoT-based smart agriculture systems. They have used advanced techniques like artificial intelligence, etc. to make instant decisions in agriculture.

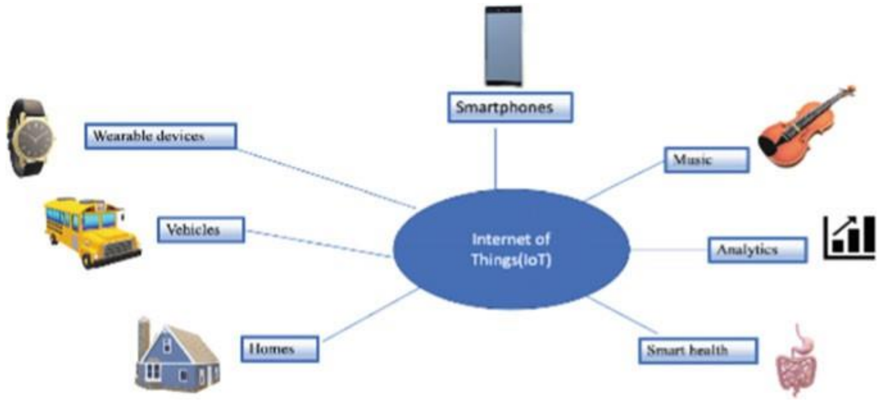


Fig. 1 Applications of IoT devices

3.1.2 IoT Architecture

The most famous architecture of IoT is a five-layer architecture given by Zhong, Zhu and Huang in [23], shown in Fig. 2. It contains five layers namely perception layer, network access layer, network transmission layer, application support layer and presentation layer. The perception layer contains devices like sensors, RFID’s, etc. Its main function is information collection. The network access layer consists of gateway and base station node. Its main function is to transfer the information to the above layers. Next is network transmission layer, which helps in transfer and exchange of information. The application support layer uses technologies like

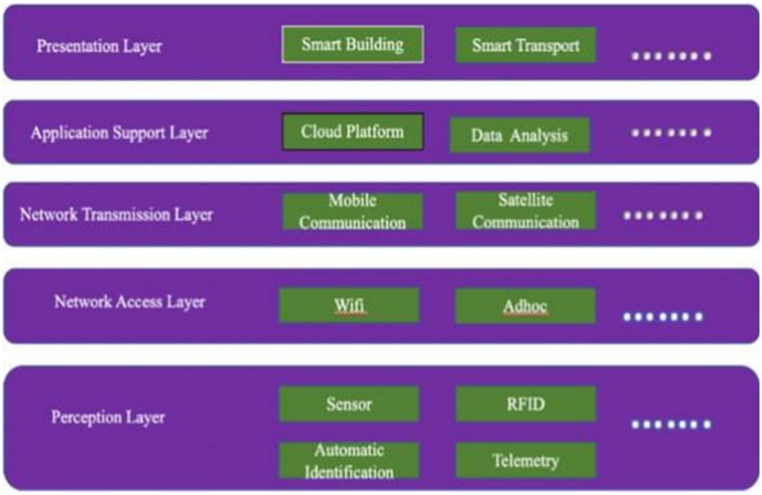


Fig. 2 Architecture of IoT

database, cloud computing, expert system, etc. to carry out information storage and analysis. Next is the presentation layer, which uses different techniques to present the intelligent information to the users.

3.1.3 IoT Communication Protocols

The popular communication protocols used in IoT system are RFID (Radio-Frequency Identification), IEEE 802.15.4, IPv6 Low-Power Wireless Personal Area Network (6LoWPAN) [24], NFC, LoraWAN, ZensysWave (*Z*-Wave) [25]. These are discussed briefly in this section.

- RFID (Radio-Frequency Identification): It is a popularly used communication technology. It helps in the monitoring of objects in real time. This technology consists of readers as well as many RFID tags [26]. The RFID tags are based on radiofrequency electromagnetic fields.
- IEEE 802.15.4: The physical layer and media access control layer contains this standard for Wireless Personal Area Networks (LR-WPANs). Its working model is based on 2.4 GHz ISM band [27].
- IPv6 Low-Power Wireless Personal Area Network (6LoWPAN): This standard specifies encapsulation mechanism. It can work on diverse communication platforms such as Ethernet, IEEE 802.15.4, Wi-Fi, etc.
- NFC (Near-Field Communication): Basically it can be seen as a communication device that works on mobiles [28]. It can transmit small amounts of data.
- LoraWAN: It is a long-range and low-power communication protocol that is best suited for widearea networks. It supports multiple applications simultaneously in an urban environment.
- ZensysWave (*Z*-Wave): It is a low power communication protocol that is well suited for small size application domains. It usually operates at 850–900 MHz bandwidth.

3.1.4 Computing Infrastructure for IoT

IoT is a system of interconnected objects or physical devices that are controlled through the internet. The objects can be anything from small chips to actual smart cars that run on the roads. The objects are fitted with sensors and actuators. The sensors collect the real-time data and give to gateways for pre-processing of data. The data is sent to the cloud for processing and data analysis. Finally, it is given to the user through user applications.

3.2 Cloud Computing

Cloud computing is a technology that provides network access to a shared collection of resources like servers, networks, applications, services, etc. on-demand. It provides more reliable access to the resources with cost savings. The characteristics features of cloud computing are resource sharing, on-demand access to services, scalability, optimization of resources, etc. There are many issues and challenges in Cloud computing. These are Security issues, interoperability issue, limited access control, handling big data, performance issues, etc. [29]. Cloud computing helps IoT applications to manage data storage and perform complex computations in an efficient manner.

3.3 Fog Computing

3.3.1 What is Fog Computing?

Cloud computing suffers from some drawbacks like increasing the bulk of data due to rise in the number of IoT devices, transferring the data to the cloud uses many resources, causes network congestion, leads to long delays, latency, etc. This gave rise to the concept of fog computing [30]. Fog computing or fogging is a layer present in between the cloud and end devices that is aimed to handle the information in an efficient manner by reducing the load on the cloud. Basically, it is augmentation of cloud computing to edge of system. Fog computing is also called edge computing as it is closer to the end clients.

3.3.2 Features of Fog Computing

Following are the features of fog computing:

- It is suited for applications that need dynamic processing of data with low latency.
- The fog nodes can be widely distributed in various geographical locations to track end users devices.
- Fog nodes can perform analysis of highly confidential data.
- Fog computing solves scalability issues that arise in cloud, by decentralising the data.
- Fog computing can manage multiple applications simultaneously.

3.3.3 General Fog Architecture

The general architecture of fog computing is shown in Fig. 3. It consists of three layers- Cloud layer, Fog computing layer (consisting of fog nodes) and users. The

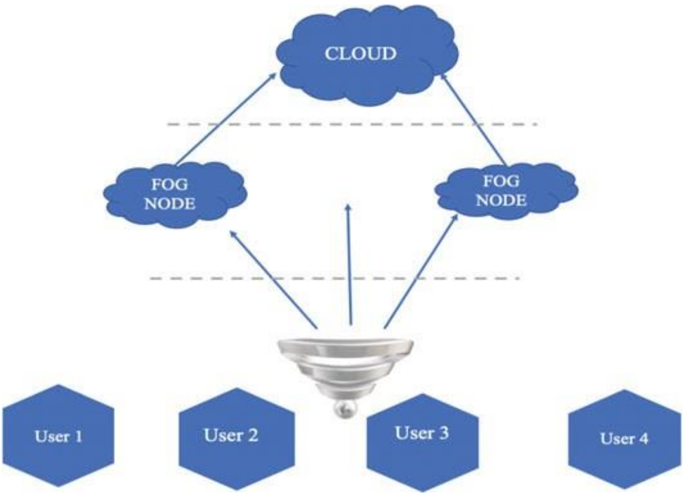


Fig. 3 Fog computing architecture

cloud layer stores information required by the applications, which are executed by the fog nodes. The fog nodes provide data privacy.

3.3.4 Hierarchical Architecture

Most of the architectures of fog computing proposed in research are hierarchical in nature. For example, Kiani et al. [31] have proposed a structure consisting of fog nodes distributed geographically and have analysed the performance. In another work, Gao et al. [32] have given a system architecture where allocation of resources is done dynamically and have proved a reduction in latency.

3.3.5 Ensemble Architecture

There is another architecture called as ensemble architecture, used for the handling of unbalanced data [33]. In this architecture, a fog cluster is deployed for a group of sensors. Then classification algorithm is applied separately to each cluster to identify class instability. The ensemble techniques are based on majority voting.

3.3.6 Hybrid Fog Architecture

There are many types of hybrid fog architectures proposed in literature. One of them is given by Suri et al. in [34]. They have proposed an architecture in which neighbouring fog nodes, at the same level are linked with each other, thereby making an auxiliary connection. The proposed architecture is implemented in a parking scenario and has proved to show lower latency than the traditional fog architecture. In another work [35], the authors have proposed a hybrid fog architecture and implemented it on a real-world problem of parking a vehicle in a shopping mall. The algorithm is able to find out the nearest free car parking slot. In [36], authors have extended the architecture given in [35] for parking problems in a single mall as well as for multiple malls.

3.3.7 Challenges in Fog Computing

Fog computing has some challenges. These are given below [37]:

- Access-related issues
- Lack of Resources
- Managing heterogeneous end devices
- Uniformity in fog computing architecture.

3.3.8 Applications of Fog Computing

Fog computing is used in many areas. Some of them are listed below:

- Smart transport system: Smart transport system consists of heterogeneous devices, having different computing capabilities. The devices are located at different geographical locations. It is a difficult task also to handle these connected devices.
- Smart health care: In the healthcare domain, particularly in emergency situations, fog computing shows better results than cloud computing due to its low latency. There are many works done in this area by different authors [38–40].
- Remote Gaming: Remote games are played from a server over the internet without any need to download it or install any software. In this, remote the game is run on the cloud and users control it from their side. To save energy and increase portability, fog computing is used.
- Smart energy management: It deals with the management of energy consumption efficiently in residential and commercial domain [41–44]. Fog computing provides good performance, customization, etc. in a reasonable cost.
- Smart agriculture: Fog computing has a prominent role in the field of agriculture such as monitoring climatic conditions and plant growth, etc. This is achieved by using smart sensor nodes.

3.4 Cloud, Fog, and IoT

This section talks about the amalgamation of Cloud, Fog and IoT and which combination is better and how.

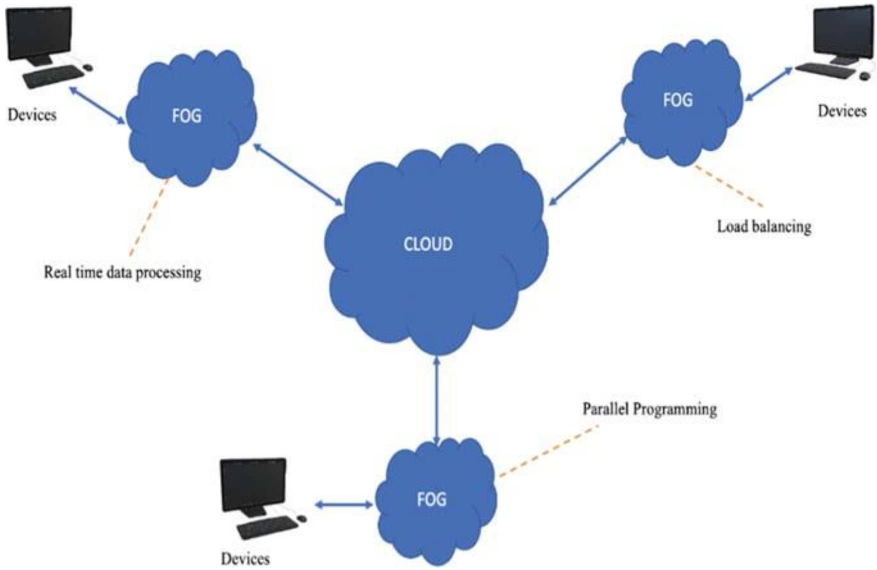


Fig. 4 Cloud-fog architecture

3.4.1 Why Fog Over Cloud?

The bulk of data generated from the devices is increasing day by day, it is becoming difficult for the cloud to handle it. Apart from this, cloud computing is also facing challenging issues like high latency, security issues, resource allocation, bandwidth issue, etc. So, new computing technologies like edge computing and fog computing have come into picture. Fog computing is a layer that is intermediate between cloud and end users/data sources. The fog nodes gather data from the devices, thereby reducing load on the cloud. This leads to low latency and more suitable for handling real-time applications efficiently. Another point is that cloud computing uses servers for data storage and computations, whereas fog computing uses edge devices [45]. Fog computing supports different applications like real-time data processing, data sensing, parallel programming, distributive load balancing and many more. The cloud-fog architecture is shown in Fig. 4.

3.4.2 Fog and IoT

Internet of Things (IoT) consists of smart and connected devices that communicate with each other. These devices are remotely located within a network. The IoT devices are used in a variety of areas like smart city, smart agriculture, smart health, smart transport and many more. These devices generate massive amounts of data regularly and also need to perform data analytics. Then Cloud computing came into picture, it uses servers for data storage and computation. The cloud computing suffered from some drawbacks like data overloading, high latency, etc. To overcome these, the concept of fog computing came into picture. Fog computing extends the concept of cloud computing by moving the resources to the end users.

4 Proposed Architecture

The aim of this chapter is to promote the use of public transport primarily, which is the network of DTC buses running on the roads across the NCT, reaching almost at every corner of the city. This would also help middle-class people not to switch to private vehicles and making them at their basic mode of transport, for daily business activities. Thereby helps in reducing the traffic congestion in the city. Besides DTC, the Delhi Metro is also working with this aim, of use of public transport as much as possible. The main drawback while commuting on DTC buses, is that if the passenger standing at the bus stop at point A wants to go to point B.

The following observations are made while waiting at the bus stop at point A,

1. He /She might not know the bus numbers, travelling on the desired route that would take him to destination point B. Then he/she will ask other passengers, standing at the bus stop for the bus numbers.
2. Also, if the bus numbers are known, then the ETA of the next desired bus is not known.

To handle this, we have proposed Hybrid Peer Connected Fog Architecture (HPCFA) for the above-mentioned problem, and which is shown in Fig. 5.

This is a hierarchical architecture composed of cloud or the web server, fog nodes and the IoT devices. The cloud or web server is at the root, lowering down the hierarchy, which is the level 1 of N-ary tree, are the fog nodes which are interconnected at the same level. The lowest level of the hierarchy are the IoT devices or the end-user devices.

The vehicles on the roads can communicate with each other using the cloud as the medium of communication and are known as Vehicle-to-Vehicle V2V [46]. In our proposed architecture HPCFA, Fog technology is also involved in addition to cloud for faster communication or reducing latency is called Vehicular Fog Computing (VFC). This HPCFA is an integration of Vehicular Cloud Computing (VCC) model and VFC, with the aims to maximum utilization of network resources. The installation of fog nodes everywhere on the roads is costly but is computationally capable with minimum latency and if we rely only on the cloud for the real-time traffic management, the cloud becomes overloaded due to terabytes of data produced by humongous vehicles. Thereby the traffic management process delays.

In HPCFA, the components are cloud, fog nodes and edge or IoT devices. The communication or the data transfer occurs in two cases:

Case I. In the absence of fog nodes, the data, which in our case is the location of the DTC bus is transmitted to cloud directly.

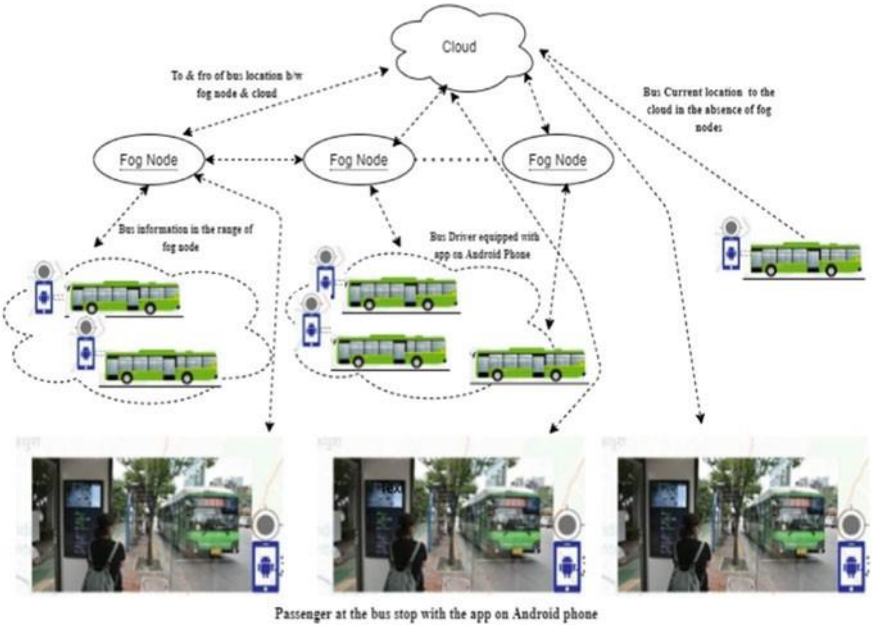


Fig. 5 Hybrid peer connected fog architecture (HPCFA)

Case II. In the presence of fog nodes the DTC bus location is sent to the respective fog node covering a certain range in distance. This information is then transmitted two-way to the peer fog node connected to it depending on the direction of the bus for tracing the location of the bus for its ETA. Further, this information is also stored in the webserver at the cloud for backup and as well as for considering the case I above.

4.1 Problem Formulation and Solution

As we are working on the public transport system of NCT, Delhi, the IoT devices are the smartphones, or the cameras and many more devices are working at the edge. In Delhi, the cameras are installed only at the prime locations-railway stations, bus stops, crowded areas, etc. but still there are many places where these cameras are not yet installed. The location of the vehicles, in our case, are the location of the buses captured by the cameras installed on the roads or by the mobile phone of the driver. The location of any vehicle is measured using the parameters latitude and longitude. This information is further communicated to the fog nodes and these fog nodes are linked to other peer fog nodes at the same level and to the cloud as shown in the HPCFA architecture given in Fig. 5.

Table 1 List of parameters included in execution

Parameters	Value
Area	4500 × 4500 m
Transmission range	5–6Kms
Number of fog nodes	4
Gain	5.8 dbi
Channel bandwidth	250 kHz
Simulation time	120 s

To show the simulation, the fog node LoRa 434 MHz, is installed whose details are shown in Table 1. Each fog node is covering the 5 km range. The DTC buses locations in their zone is respectively updated in this local cloud. The number of buses available in that area can be retrieved from the respective fog node and the information is sent to the user. As shown in Fig. 5, these fog nodes are interconnected and can transfer the information bidirectionally. Every fog node updates its database at every 3 ms and the database in the user's mobile phone app will also be updated after every 3 ms. Since these fog nodes are interconnected, therefore when a bus comes out of the ranges fog node say f1 then its last location will be communicated to the peer fog node say f2. In this way the path of the bus can be traced from fog node f1 to next fog node f2 and so on. This hopping of information from one fog node f1 to other connected node f2 in 5 ms. Further hopping of information from the fog nodes to the cloud is updated after every 10 ms.

The response time (R) for getting the actual location of the bus—the time required to transfer the of information from the bus to the fog node (T_1), from the fog node to the user (T_2) and from a fog node to the peer fog node(T_3).as shown in Eq. 1.

$$\mathrm{R} = \mathrm{T1} + \mathrm{T2} + \mathrm{T3} \tag{1}$$

As shown above the response time with fog nodes is always less as compare to the information of the bus location from the cloud to the user. Therefore HPCFA, is energy efficient in terms of time as compared to the traditional architecture.

The Table 1 shows the list of parameters that we included in our execution of proposal.

A database is created having all information about each DTC bus like departure time, Arrival time, number of stops, current location, Fare. Once, the user sets the source and destination place on the application a request is generated having user location in form of Latitude and longitude coordinates like **28.644800, 77.216721**. These latitude and longitude coordinate information are further decoded by the application into addressable form (means the places in Delhi like Karol Bagh, Azadpur) and sent to the database. Database fetches the destination point from the request and will check if any bus available on that route or not if yes, then the application algorithm starts comparing the distance between the available buses and the users. And show only those bus's which are nearby to the user.

The proposed architecture will handle both the cases to track the real-time location of the bus, so that the ETA of the bus can be notified to the passenger standing at the bus stop.

Steps involved for using this application:

1. Authentication
2. Allow mobile location
3. Enter source
4. Enter destination
5. Get Available buses
6. Users can see the current bus location.

The architecture is further explained with the help of a flowchart given in Fig. 6.

5 Simulation and Results

To show the simulation and results, we have designed an android application for our proposed architecture. The Fig. 7 (from 7a to f) shows the interface.

The user needs to authenticate himself using his mobile number and turning on the live location. Once done, the application asks for source which he can enter manually or just click on the get location option to get the current location. Next, he needs to enter his destination and by clicking proceed the list of bus option(s) available for him will be displayed on the app. The user will get the ETA of the bus to his bus stop along with live tracking of the bus. Additional information like bus fare, bus route and estimated time to reach his destination is also mentioned. After seating in preferred bus the user gets features like next stop details and estimated time required to reach his destination. A panic button for use in emergency situations is also provided which sends the current location to the nearest police station.

6 Conclusion

In this chapter, we have tried to solve real-time transport-related problems. An architecture called as Hybrid Peer Connected Fog Architecture (HPCFA) is proposed to lower the latency time overhead. In HPCFA, the fog nodes are organized in a hierarchy in which the peer fog nodes at the same level are interconnected with each other. The proposed architecture is very flexible, as it works both with fog nodes or without fog nodes and directly with the cloud. It is also simulated using fog node LoRa 434 MHz which covers a range of 5kms. The proposed architecture HPCFA is a hybrid architecture and the response time of delivering the actual location of the bus is less as compared to the transfer and retrieval of information from the cloud. Therefore, our proposed architecture is also energy efficient and less expensive. The

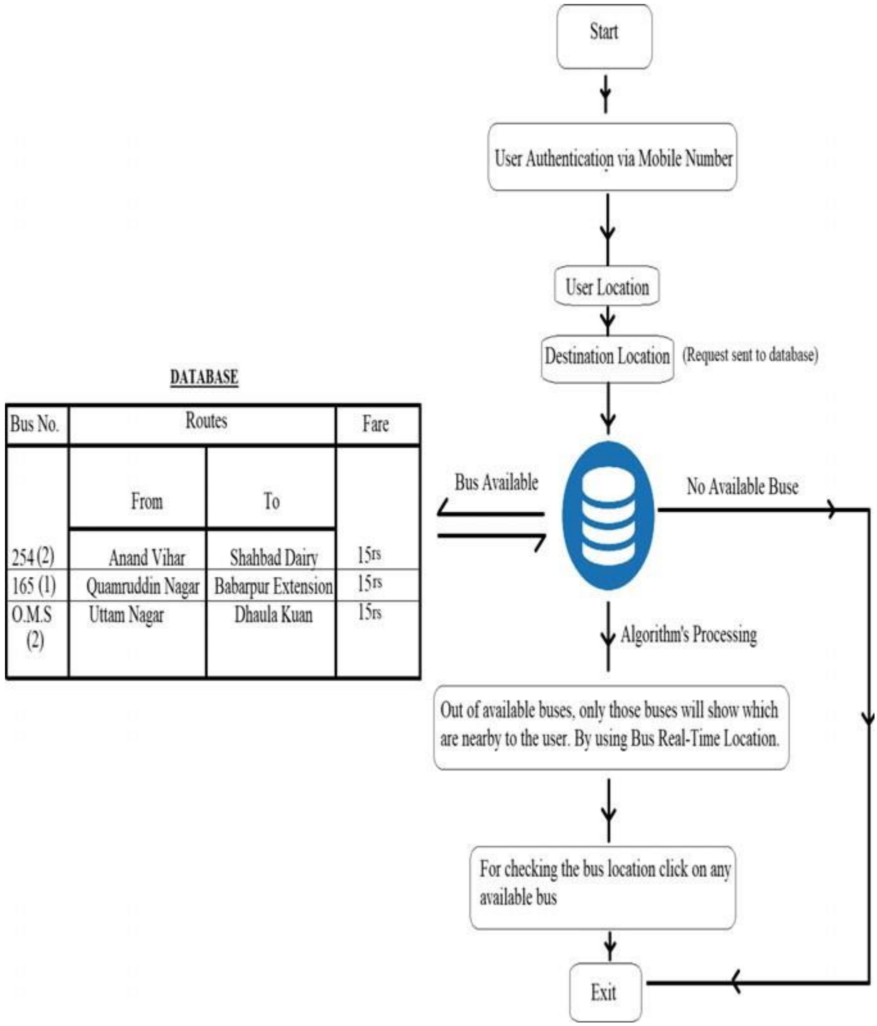


Fig. 6 Flowchart

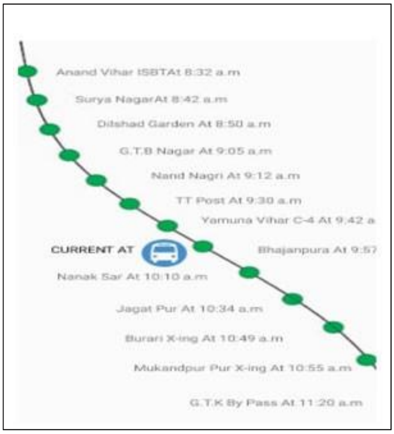
results of the simulation are shown with the help of Android application of the user where the user gets the ETA of the next bus and its actual location in his smartphone.

a. Simulation-1

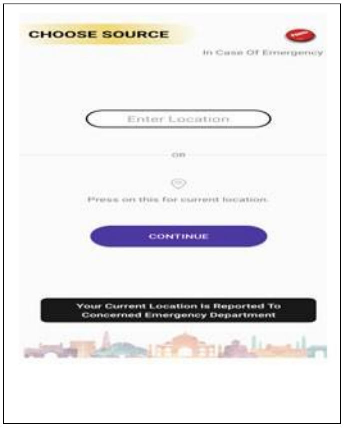
b. Simulation-2

c. Simulation-3

d. Simulation-4



e. Simulation-5



f. Simulation-6

Fig. 7 Simulations

References

1. Gharaibeh A et al (2017) Smart cities: a survey on data management, security, and enabling technologies. IEEE Commun Surv Tutorials 19(4):2456–2501
2. Bhatnagar H. (2018) Smart transportation: a study of the transportation sector reforms and developments in India, Synergy. ITS J ITT Manage 16(1):60–72
3. Dash DK Delhi traffic costs Rs 60,000 crore annually. The Economic Times. https://economictimes.indiatimes.com/news/politics-and-nation/delhi-traffic-chaos-costs-rs-60000-crore-annually/articleshow/56980296.cms. Last Accessed on 5 Dec 2020
4. Humphries M, Radev P, Shirvaikar M (March 2005) A realtime vehicle tracking system. In: Proceedings of the IEEE Thirty-Seventh Southeastern Symposium on System Theory, SSST'05, pp. 357–361
5. Abdelhamid S, Hassanein HS, Takahara G (2015) Vehicle as a resource (VaaR). IEEE Netw 29(1):12–17
6. Miranda WM et al (2017) BusMe: automatic bus localization system and route registration. Procedia Comput Sci 109:1098–1103
7. Curzel JL, Lüders R, de Morais GP (2016) On modeling interleaved events in a bus transportation system with real-world data monitoring. IFAC-PapersOnLine 49(3):203–208
8. Sonawane A, Bhanushali A, Gogri K, Khairnar M (2020) Real time bus tracking system. Int J Eng Res Technol (IJERT) 9(06), ISSN: 2278–0181
9. Shoeb M, Ahmed J (2017) Tracking of DTC bus using GPS and Android. Int Res J Eng Technol (IRJET) 04(04)
10. Awad AY, Mohan S (2019) Internet of things for a smart transportation system. Int J Interdisc Telecommun Networking (IJITN) 11(1):57–70
11. Huang L (March 2010) Real-time multi-vehicle detection and sub-feature-based tracking for traffic surveillance systems. In: 2010 2nd IEEE international Asia conference on informatics in control, automation and robotics (CAR 2010), vol 2, pp 324–328
12. Wu H, Meng Y (July 2019) A novel multi-vehicle tracking method based on linked list of associated vehicles. In: 2019 IEEE 4th international conference on image, vision and computing (ICIVC), pp 376–381
13. Ning Z, Huang J, Wang X (2019) Vehicular fog computing: enabling real-time traffic management for smart cities. IEEE Wirel Commun 26(1):87–93
14. Ma R et al (July 2020) Energy efficient software matching in distributed vehicular fog based architecture with cloud and fixed fog nodes. In: 2020 IEEE 22nd international conference on transparent optical networks (ICTON), IEEE, pp 1–4
15. Hou X et al (2016) Vehicular fog computing: a viewpoint of vehicles as the infrastructures. IEEE Trans Veh Technol 65(6):3860–3873
16. Sutagundar A, Attar AH, Patil B (July 2018) Resource allocation for fog enhanced vehicular services (FEVS). In: 2018 IEEE international conference on inventive research in computing applications (ICIRCA), pp 360–365
17. Mehta M, Patel K (2020) A review for IOT authentication–current research trends and open challenges. Mater Today Proc
18. Dr Taneja S, Chandna M (2020) IoT for diabetes: smart monitoring and control. In: Proceedings of 3rd international conference on computing informatics and networks (ICCIN), Springer LNNS series 2021(under publication)
19. Ellaji C, Sreehitha G, Devi BL (2020) Efficient health care systems using intelligent things using NB-IoT. Mater Today Proc
20. Debauche O, Mahmoudi S, Manneback P, Assila A (2019) Fog IoT for Health: a new architecture for patients and elderly monitoring. Procedia Comput Sci 160:289–297
21. Ratnaparkhi S et al (2020) Smart agriculture sensors in IOT: a review. Mater TodayProc
22. Bu F, Wang X (2019) A smart agriculture IoT system based on deep reinforcement learning. Futur Gener Comput Syst 99:500–507

23. Zhong CL, Zhu Z, Huang R (Aug 2015) GStudy on the IOT architecture and gateway technology. In: 2015 14th IEEE international symposium on distributed computing and applications for business engineering and science (DCABES), pp 196–199
24. Khanna A, Kaur S (2020) Internet of things (IoT), applications and challenges: a comprehensive review. Wirel Pers Commun 114:1687–1762
25. Zaidan A et al (2018) A survey on communication components for IoT-based technologies in smart homes. Telecommun Syst 69(1):1–25
26. Dominikus S, Aigner M, Kraxberger S (2010) Passive RFID technology for the internet of things. In: IEEE international conference for internet technology and secured transactions (ICITST), pp 1–8
27. Jiang W (2015) A diagnostic tool for the causes of packet corruption in wireless sensor networks. Master's thesis, Department of Information and Communication systems, Mid Sweden University
28. He D, Kumar N, Lee J-H (2015) Secure pseudonym-based near field communication protocol for the consumer Internet of things. IEEE Trans Consum Electron 61(1):56–62
29. Zhang Q, Cheng L, Boutaba R (2010) Cloud computing: state-of-the-art and research challenges. J internet serv Appl 1(1):7–18
30. Hong HJ (Dec 2017) From cloud computing to fog computing: unleash the power of edge and end devices. In: 2017 IEEE international conference on cloud computing technology and science (CloudCom), pp 331–334
31. Kiani A, Ansari N, Khreishah A (2019) Hierarchical capacity provisioning for fog computing. IEEE/ACM Trans Netw 27(3):962–971
32. Gao X et al (2019) Pora: predictive offloading and resource allocation in dynamic fog computing systems. IEEE Internet Things J 7:72–87
33. Joo Y, Lee J, Jeong J (2020) Ensemble fog computing architecture for unstable state detection of hydraulic system. Procedia Comput Sci 175:230–236
34. Suri B et al (2019) Peering through the fog: an inter-fog communication approach for computing environment. In: International conference on innovative computing and communications. Springer, Singapore, pp 73–81
35. Suri B, Pramanik PKD, Taneja S (2020) An efficient parking solution for shopping malls using hybrid fog architecture. In: International conference on innovative computing and communications. Springer, Singapore, pp 313–320
36. Suri B, Pramanik PKD, Taneja S (2019) A hybrid fog architecture: improving the efficiency in IoT-based smart parking systems. Recent Adv Comput Sci Commun
37. Mukherjee M, Shu L, Wang D (2018) Survey of fog computing: fundamental, network applications, and research challenges. IEEE Commun Surv Tutorials 20(3):1826–1857
38. Zao JK et al (Jun 2014) Augmented brain computer interaction based on fog computing and linked data. In: Proceeding IEEE international conference intelligent environment. Shanghai, China, pp 374–377
39. Aazam M, Huh E.-N (March 2015) E-HAMC: leveraging fog computing for emergency alert service. In: Proceeding IEEE international conference pervasive computing communication Worksshops (PerCom Workshops), St. Louis, MO, USA, pp 518–523
40. Preden JS et al (2015) The benefits of self-awareness and attention in fog and mist computing. Computer 48(7):37–45
41. Singh R, Gehlot A, Khilrani JK, Mittal M (2020) Internet of things–triggered and power-efficient smart pedometer algorithm for intelligent wearable devices. In: Wearable and implantable medical devices. Academic Press, pp 1–23
42. Singh R, Gehlot A, Mittal M, Samkaria R, Choudhury S (2017) Application of icloud and wireless sensor network in environmental parameter analysis. Int J Sens Wireless Commun Control 7(3):170–177
43. Mittal M, Pandey SC (2019) The rudiments of energy conservation and IoT. Energy conservation for IoT devices. Springer, Singapore, pp 1–17
44. Singh PP, Khosla PK, Mittal M (2019) Energy conservation in IoT-Based smart home and its automation. Energy conservation for IoT devices. Springer, Singapore, pp 155–177

45. Baucas MJ, Spachos P (2020) Using cloud and fog computing for large scale Iot-based urban sound classification. In: Simulation modelling practice and theory, vol 101, pp 102013
46. Mohmmad S, Sheshikala M, Shabana (2018) Software defined security (SDSec): reliable centralized security system to decentralized applications in SDN and their challenges. J Adv Res Dyn Control Syst 10(10)-Special Issue, pp 147–152

Analysis on Application of Fog Computing in Industry 4.0 and Smart Cities

Suja Cherukullapurath Mana, B. Keerthi Samhitha, D. Deepa, and R. Vignesh

Abstract In this chapter, authors analyzes how fog computing can be efficiently utilized to improve the productivity in the industry 4.0 and smart city applications. The main aim of industry 4.0 applications is to improve the efficiency of manufacturing process through the incorporation of latest technologies. This environment can be improved by incorporating the fog computing paradigm. High energy consumption and abundance of data to be processed at the data nodes are some of the challenges that need to be addressed in industry 4.0 and smart city implementations. A fog computing-enabled architecture helps to reduce some of these challenges by working as a low complexity computational layer between cloud and internet of things (IoT) layers. By introducing this fog layer computationally intensive data processing tasks can be moved from the cloud layer to the fog layer and this fog layer can also act a gateway to the other upper layers. In smart city applications also fog computing can be effectively utilized. In the fog computing environment the data analytics tasks can be pushed to the edge of the network which leads to better efficiency. In fog computing paradigm major functionalities are moved near to the local nodes. Since most of the computations are happening locally the need of transferring data to the cloud servers is significantly reduced. The benefits of this architecture can be utilized to improve the efficiency of smart city and industry 4.0 implementations. In this chapters, authors analyze how fog computing can be effectively utilized to improve industry 4.0 and smart city applications.

Keywords Fog computing · Industry 4.0 · Smart cities

1 Introduction

Fog computing is a paradigm in which potential functionalities are placed at the edge of the network. The main objective of fog computing is to have significant amount of communication happening locally. These fog nodes are collocated near to the data

S. C. Mana (✉) B. Keerthi Samhitha D. Deepa R..Vignesh
Department of Computer Science and Engineering,
Sathyabama Institute of Science and Technology, Chennai, India

R. Tiwari et al. (eds.), *Energy Conservation Solutions for Fog-Edge Computing Paradigms*, Lecture Notes on Data Engineering and Communications Technologies 74,
https://doi.org/10.1007/978-981-16-3448-2_5

generating devices so that significant amount of communications can be performed locally without sending too much data to the cloud network servers. Fog computing works on the concept of proximity to the client. This architecture results in significant savings in bandwidth and reduction in network latency. Fog architecture can be effectively implemented in internet of things (IOT). In internet of things millions of devices are connected together and transfer data with each other. Introduction of fog computing will result in significant amount of savings in the internet of things enabled environment. Fog computing is defined as a recourse paradigm resides in between the local nodes and the data servers. It can also be considered as an extension of cloud computing to local nodes. Since most computation can be done locally without sending the data to the cloud server through the network, major savings are achieved in terms of network bandwidth and latency.

As internet of things applications become prevailing the amount of devices and sensors being used are huge. Each sensor is sending huge amount of data to the network creating a lot of network traffic. It will consume a lot of network bandwidth also. The fog computing paradigm is implemented in such a way that the computations are located near to the client nodes. Since major functionalities can be performed locally the need of consuming network bandwidth will be greatly reduced. The prevalence of internet of things also leads to the applications like smart cities and technology-enabled manufacturing. In both industry 4.0 and smart cities we are using iot connected devices with a huge volume of sensors. The data that is being generated and transmitted in these applications are of large volume. Today's cloud architecture is not sufficient to accommodate the volume, velocity, and varieties of such implementations. Also connecting this huge amount of applications directly to the cloud impart some practical difficulties. Some of the challenges are high latency, requirement of high bandwidth, security concerns, reliability, and scalability. Fog computing architecture can address some of these challenges. A diagrammatical representation of fog computing paradigm is given below in Fig. 1.

A diagrammatic representation of cloud architecture is given in Fig. 1. As shown in the diagram the fog nodes bring the computational functionalities near to the data nodes. Most of the computational requirements can be satisfied by the fog nodes itself. So the amount of traffic from the data nodes to the cloud servers can be greatly reduced. It results in greater bandwidth savings. Fog computing paradigm can be effectively utilized in industry 4.0 applications and smart city implementations. Basically, fog computing paradigm can be considered when a huge volume of things at the network endpoints are generating huge volume of data and the latency is a primary concern of any network.

Industry 4.0 is a scenario in which the capabilities of artificial intelligence and cloud computing are being utilized in the manufacturing sector. It aims to make manufacturing smart by incorporating the capabilities of artificial intelligence and automation. It applies the development in latest technologies to the manufacturing field. Industry 4.0 applications work based on the concept of automation, real-time data, and artificial intelligence. It incorporates data analytics and machine intelligence into the manufacturing field. Fog computing can facilitate the industry 4.0 applications by bringing the cloud functionalities close to the local node. The efficiency

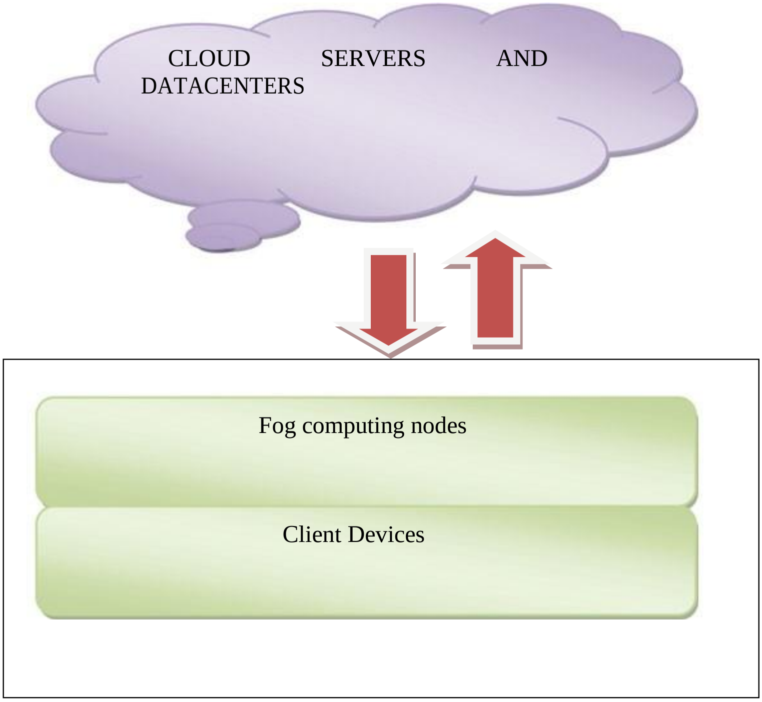


Fig. 1 A representation of cloud architecture

of industry 4.0 applications can be enhanced by incorporating the fog architecture into it. By incorporating the fog architecture the capabilities of cloud computing will be extended to the edge of the network and it will positively influence the latency and reliability of industry 4.0 implementations. Fog computing enabled industry 4.0 applications are diagrammatically represented in Fig. 2.

Smart city is an internet of things enabled city. Smart cities are characterized by smart homes, smart transportation, smart healthcare, smart waste management, etc. Some components of a smart city implementation is given in Fig. 3. Internet of things technology plays an important role in this smart city implementation. In this implementation also many computationally intensive devices and sensors are being used. These sensors and devices will transfer a lot of data and consume lot of energy and network bandwidth. Implementing the fog architecture facilitate the smart city implementation by increasing the efficiency. Fog architecture brings the cloud capability to near the local nodes in smart city applications so that the data processing can be made easy and network delay can be reduced to a great extend.

This chapter progresses in such a way that in upcoming sections authors discuss about fog computing architecture in detail and also discuss about application of fog computing in industry 4.0 and smart cities. The advantages and challenges of such implementations also discussed in detail.

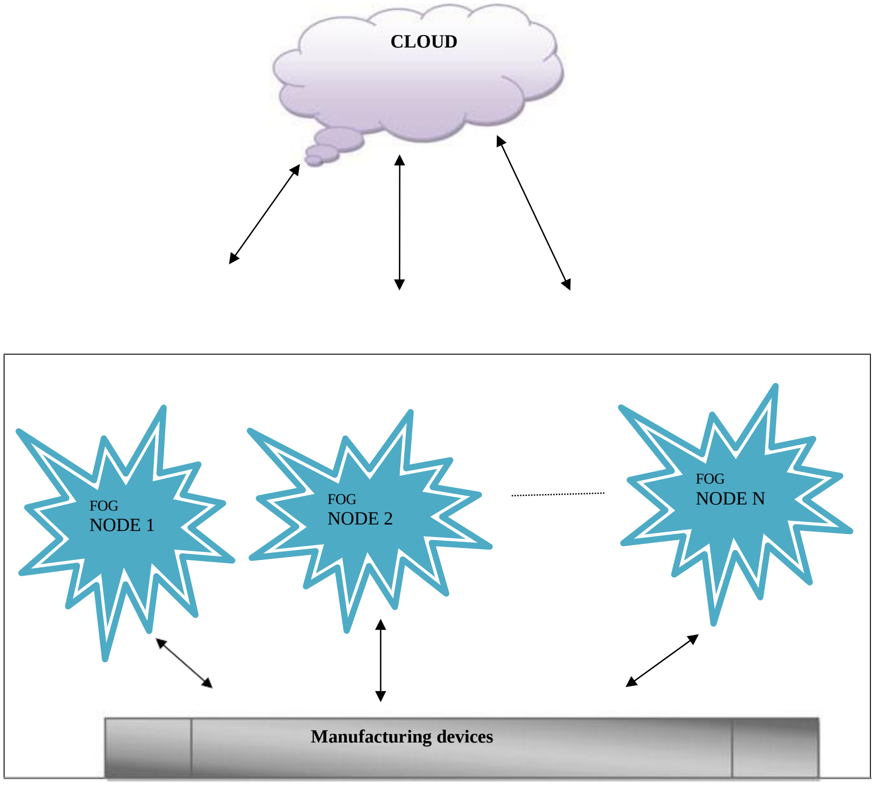
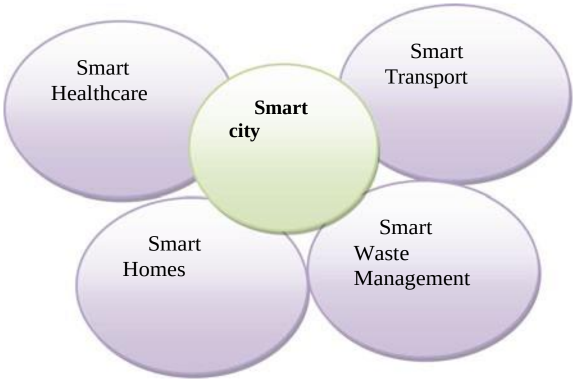


Fig. 2 Fog computing enabled industry 4.0

Fig. 3 Some components of a smart city



2 Overview of Fog Computing

Fog computing is a paradigm in which the near user devices are used to perform the computation. It allows the incorporation of many parts and services. The overview of fog computing is based on distributed computing foundation in which data, computation, storage, and different types of applications are situated subsequently among the data source and the cloud. Likewise, edge and fog computing get the dominance and potential about cloud imminent into where the data has been generated and responded. So many users are using the details about edge and fog computing reciprocally as a result of both denotes attend processing and intelligence close to where the data has been generated. The indicated part usually completed through better ability, still this also may be completed because of protection as well as reasons for compliance. This approach of complicated fabric extends upon outside edges of where the actual data is generated and where this will be finally stored. It will check whether it is present in cloud or in customer's data center. This will be another layer of a shared network environment. It is nearly included under cloud computing as well as internet of things (IoT). The edge of the network is where data from IoT devices is created. Essentially, the advancement of fog computing groundwork provides configuration on many options as long as transformation of data everywhere. It is more applicable for doing new configurations (Fig. 4).

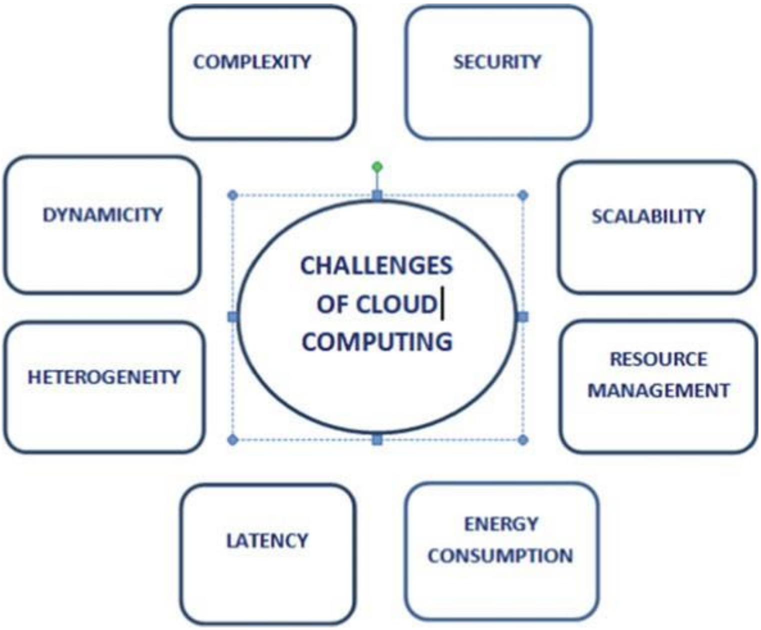


Fig. 4 Challenges of cloud computing

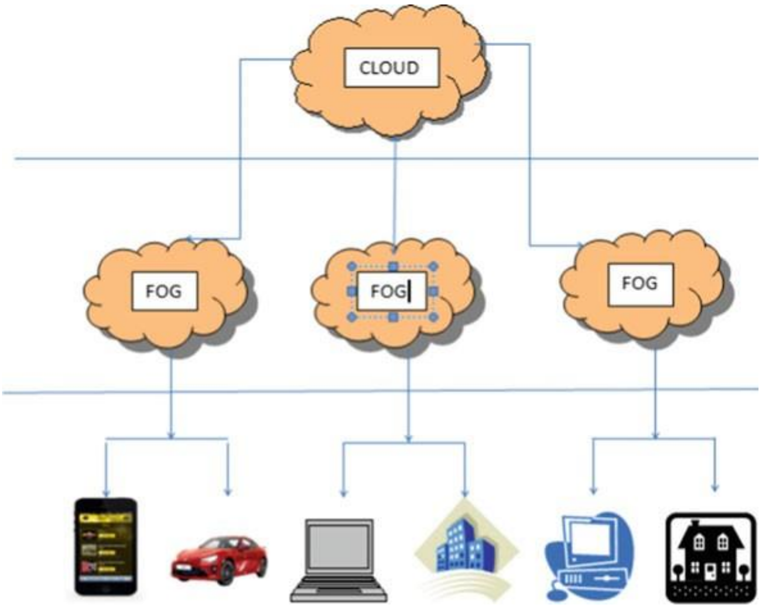


Fig. 5 Fog computing architecture

Fog computing generates that low-latency framework connections among all the machines as well as analytical end devices. The framework structure depreciates the supply among few bandwidths have to be correlated with the information need to be posted with all the information collected from the data center or cloud. This can further be used in the applications everywhere in there is no other bandwidth connection needs to be sent data, so this will be continued close upon where the data is generated. As we include that, people can use protected models in the fog network, which is disconnected network traffic to virtual firewalls to be secured. Figure 5 shows an architecture of the fog computing.

The applications of fog computing are the formless division upon actuality coiled under the express categorizations and there are many varieties of use cases, they have to be analyzed as imaginable flawless models or applications for fog computing.

When the fog computing is connected to cars

The appearance of independent as well as self-driving cars will be greater in amount with the previously huge amount of information for vehicles generate. Cars can control the operation of semi-autonomous demands the capacity with the nearly consider the positive dataset in problem-solving time follows circumstances, conditions of the driving as well as directions. Information might be sent backward to the producer to provide guidance for better vehicle maintaining or detect the usage of the vehicle. The fog computing framework will be facilitated information for the source of the data and combined at the edge computing (inside the vehicle-car), and to the producer (end point).

When it is used in smart cities and smart grids

Likewise all the connected vehicles, service frameworks are used in increased manner and problem-solving datasets are very accurately running all the systems. Frequently the data is based on the rural areas, then functioning near to where they are generated. Fog computing structure will enable for to clear the result for these issues.

When it is used in real-time applications

With the producing frameworks that have to be needed to react the given applications as they need to give the result, for the finance organizations the usage of all the real-time techniques are to give the details about decisions for trading or fraud detection.

3 Overview of Industry 4.0 and Smart Cities

Industry 4.0 gives the information for newly Industrial Reformation. Firstly that implies the advancement in manufacturer as well as functional systems with the new digital technological world. The creation of a digital activity that will not be only be connected and independently, this will be informed, investigated. The usage of information to the drive for another brilliant process is to be returning to the physical world.

This implies the many ways in smart, joined technical world will be as embedded to many places, users, and belongings, and it is represented with the capacities follows as robotics, data analytics, deep learning / artificial intelligence and nanotechnology, quantum computing, the Internet of Things, additive manufacturing, and many more advanced things.

The industry 4.0 also known as Industrial revolution, breakthrough interchange of the way business operation as well as by development, the post by whichever they are affected to go after the technology. Configurations should be decided where and how to save in these new technology and to find whichever they may be the better one to meet their needed things. Externally the forbearing among the exchanges and openings of Industry 4.0 gives the organizations risk lose platform.

The economy leadership people confirmed with the actual information and communicated networks, the transfer to problem-solving accessed to data and intelligent facilitated by the Industry 4.0 will be essentially changed to the way actual business is conducted. Combination of digital data with various resources as well as areas can drive the physical world by forming the business, in an ongoing order.

This will change the companies and their models, frameworks, and processes. We can go through few points that will be used by business leaders for to see the various situations in whichever the industry 4.0 can be affected the respective situations.

The digital world of processes, productions, network, and manufacturers enabled organizations to join the training from users, systems, analytical needs, and prediction of the insights with the better and many real decisions.

This is the main method to understanding purpose, because it's not only the "manufacturers"—it can include all of us.

The fourth industrial revolution that has been increased with many business systems, the work place, and society itself, it is conducted by supplies and manufacturer compose the main thing of the physical world. How and what the systems are made, where they are made of and how they bring to the users, and where will they go, all these information are based on manufacturing the life cycle (Fig. 6).

Smart city is an implementation in which people are enabled with smart homes, smart healthcare, smart transportation, smart energy management, smart waste management, etc. Smart home is the one in which all devices will be connected with internet of things enabled devices, so that home will be automated. Remote control of home devices is possible. Several sensors are being installed to collect and monitor data. Smart healthcare also is an essential component of smart city. Sick people are always being monitored by handheld devices attached to their body and they are being monitored even if they are away from the hospital. Smart waste

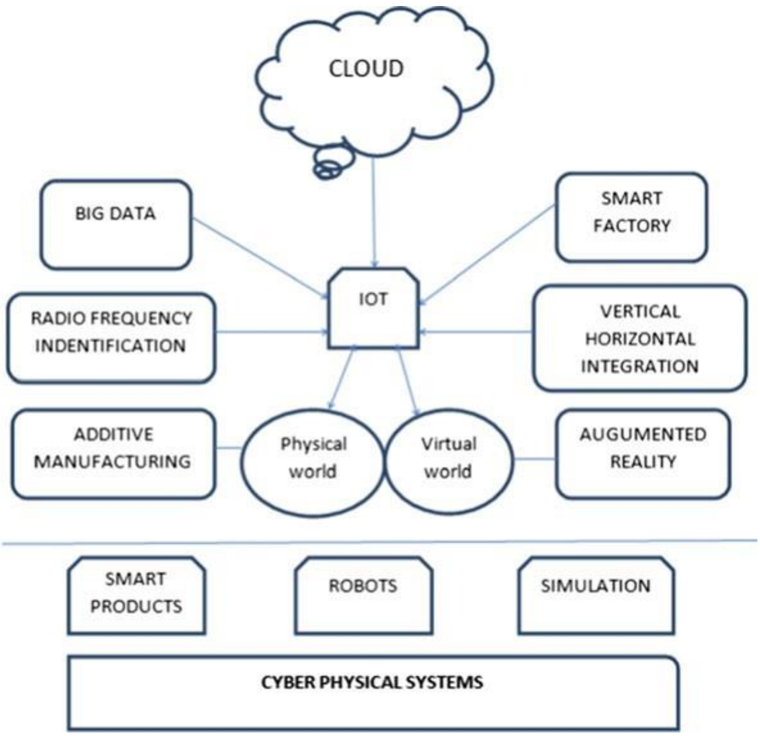


Fig. 6 Industry 4.0 architecture

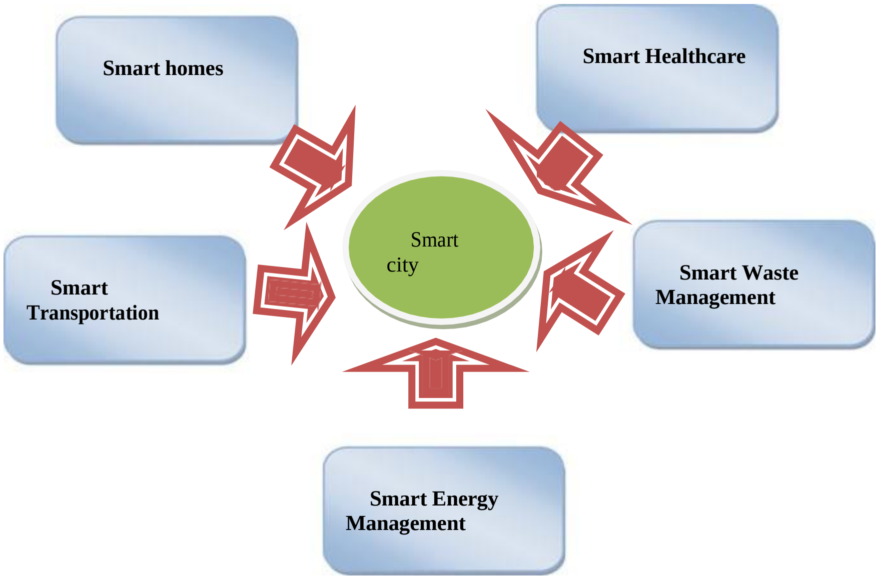


Fig. 7 Components of a smart city

management, smart energy management also are essential components of a smart city implementation. Figure 7 shows a diagram of smart city components.

4 Related Studies

Mahmud et al. proposed context-aware policy for fog computing environments. To minimize the service delivery between IoT devices along with fog nodes such as for navigation applications. It ensures the flowing of the input data towards the fog nodes overhead and application network congest significantly. This performance is evaluated both real and simulated environments and compared with the older policy. And they getting 16% of the improved service latency, computing management over the other policies [1].

Similarly Lin et al. [2] discussed in the paper about intelligent computing which consists of fog computing, cloud center, edge,gateways and sensors to facilitates all the center logistics system. Also, they developed integer programming model for fog devices, gateways, and edge nodes with their respective potential sites, from that cost is minimized under some criteria based on demand capacity, time latency, and maximum capacity of devices. Also, they proposed NP-hard facility for location-aware algorithm which incorporates discrete solutions for computational efficiency.

Recently, Bouzarkouna et al. investigates industries coverage of several components of IoT, cloud, bigdata, etc. All difficulties focused towards integration of data management in Industry 4.0 because of challenging data gathering from IoT devices. With help of Fog computing it tries to solve this kind of problem in industry 4.0. In this paper, authors explaining the role of fog computing solutions in enhancing cloud layers and the manufacturing systems. Challenges preventing implementation manufacturing are well studied and justified [3].

Verba et al. discussed graph analysis of fog computing systems concerns. Industry 4.0 fog computing recommendation and industry asset software-defined networks(SDN) used for visualization framework with different hubs. They have proposed use case scenario to determine graph-based system that will use for scale and much more realistic testing environment compared with other networks. They have shown key graph system with comparing with all other nodes [4].

Fraga-Lamas et al. [5] proposed based on Bluetooth 5 ICPS Navantia's pipeline methodology. To validate ICPS Navantia's pipe is modeled for 3D radio planning simulators which are deployed in Bluetooth 5 fog computing nodes. They got results and comparing with radio results to the simulation results and both are close enough through measurements. Though the simulation deployment time is reduced with the complex industrial appplications in bluetooth 5 fog computing system.

Similarly, Hernandes et al. proposed an architecture of smart cities and uses of objects in urban context which will help for offering services through all the cities. It is possible to implement smart cities concepts successfully. Along with that, it provides emergency services calls like police,firefighters. Some of emergency situation, the most appropriate mobile units may be required. Therefore, a middleware can be used to support the activation of the most suitable mobile units. The bigger question is which is the vantage of choice the most suitable mobile unit compared to the choice of random. In this way this paper presents a new event model and a middleware called Event to Best Subscribers (E2BS) aim to answer this question through two use cases [6].

Javadzadeh et al. in their article aim to provide smart city overview and systematic review on fog computing trending applications. Also many analytical comparison in survey are discussed in this paper. In their point of view in future reaserch, more attention should be paid to the scalability of fog computing applications in different domains. According to them, data distribution, security, and privacy are some of the important directions for future research, especially in healthcare systems. Moreover, the fog computing solutions that support 5G should be in the center of attention. In addition, handling big data analytics in fog computing applications is an open issue [7].

Zhang et al. [8] proposed in the paper the concept of IoT based on fog computing system. This article effectively deals with solutions towards with big size of data processing along with scalability. For user security and privacy problems, they proposed hierarchical network architecture. For realization and effectively solving problem in smart cities, they proposed and analyzed layered fog computing-based IoT architecture for solving the problems of big data processing and network scalability [8].

Furthermore, Singh et al. have done an analysis of energy consumption in fog computing approches in smart cities. This article proposed a secured architecture with BFAN(blockchain and fog computing network) for Internet of Everything (IOE) application of smart cities. Aim of this method architecture is to reduce latency and improving security features in blockchain technology [9].

Zahmatkesh et al. highlighted in this paper future reasearch challenges based on problems and issues related to fog computing environment. In this paper authors discussed about fog computing technologies and applications towards smart cities based on IoT environment. Along with benefits and technique use of Machine Learning, Artificial Intelligence and Unmanned Aerial Vehicles based on fog based IoT systems [10].

Relatively Badidi et al. addresses idea of smart cities towards big data implementation and analyzing challenges and other parametrics. Smart cities take new steps migration techniques from information technology to utility supply technology systems. Many smart application based on time-sensitive and quick analyze data for various emerging fog and edge computing which addresses the issues like latency and network bandwith with various big data storage and adopting emerging fields [11].

Gia et al. present in this paper based on hybrid crossover edge/fog distributed computing engineering for observing climate boundaries and traffic stream in a city with a little impression as far as introduced foundation. Specifically, we put an accentuation on traffic observing and proposed a lightweight picture preparing calculation that appraisals traffic density [12]. Similarly Caizaet al. taken review on fog computing industrial applications in the cloud with the requirements like reliability and latency checking with all other monitoring system. They have done review and challenges related to industrial internet of things with the current infrastructure-related works. Fog computing related to the industry 4.0 new paradigm with the growing new opportunity and its challenges. With the help of new hetrogeneous architecture require to obtain and delivery edge service along with network capacity. This research systematic review most related to security, fog architecture and energy consumption, and industrial level of challenges of this new technology [13].

Similarly, Khattak et al. in this paper discussed related limitation of cloud and systematic litrature review(SLR) approches in fog computing taxonoies. Further classified the proposed approches towards smart city development and concern towards solving the various challenges and various perspective into the account. Challenges and issues are classified subclasses and solving hybrid mode of comprehensive method system [14]. Relatively Vignesh et al. discussed fewer deployment in the cloud storage with low cost replication, higher availability with better performance in geo-replicated system by data centers with these benefits [15]. Deepa et al. proposed an idea of detecting road damage by image processing in smart phone and sending the co-ordinate point to cloud from cloud user can visualize the road where the damages there it will shows in map. From this users can able to avoid the accidents, etc. [16].

Samhitha et al. [17] proposed using convolutional neural network classify tumors. And also talks about CNN to identify tumors that are dangerous in lung disease. The CNN technique has the lot features and standard representation pneumonic radiological complexity, fluctuation, and classification of lung nodule.

Nelson et al. aims to synthesis and existing analysis on big data driven applications. Using and replicating data from BDO, CompTIA, Deloitte, ISC, KPMG, Nutter Consulting, PTI, SSC, and USDN, we performed analyses and made estimates regarding Internet of Things devices and environmentally sustainable urban development. Data were analyzed using structural equation modeling [18].

Kiwelekar et al. sharing resource knowledge towards distributed fog computing. Cloud and fog computing shares the properties like mobility, reduced latency, and location-aware functionalities. As well in this chapter describes about block chain technology which supports decentralized cryptographic preserving protocols and smart contracts. Along with security policy for authentication and communication trust management in fog services [19] are also important. Similarly, applications which uses the technologies of cloud computing are discussed in various papers [20–22]. Fog computing arises as an extension of this cloud architecture closes to the working node [21].

Similarly, Singh et al. they propose a deep Learning-based on IoT-oriented infrastructure for secure smart cities for challenges towards latency, cost, and centralization of data processing in smart cities. Software-Defined Networking (SDN) distributed environment for establishing protocols in network forwarding data. They compare their results with existing quantitative analysis and other parameters from that they obtained improved performance result [23]. Malik et.al Fog computing is a distributed technology which performs computations at the edge of the network in order to collect data from IoT devices generating heterogeneous data. To handle this kind of allocation issue fault tolerance and time-sensitive issue they have developed smoother and quick processing decision-making in all over smart cities [24].

5 A Fog Computing Enabled Smart City

A smart city is one in which people are equipped with smart homes, smart healthcare, smart transportations, etc. All the day today activities are being aided or supported by iot-enabled devices. These devices have a lot of sensors collect and handle huge volume of data. These data need to be analyzed and processed to generate insights and facilitate the smart city application. A cloud processing center is necessary to ensure ubiquitous computation. The data collected needed to be sent back and forth from the cloud to the edge devices. This will create a lot of network traffic and result in latency. Some other problems also arises in this implementation which is discussed below.

Scalability—As the system scales the number of sensors and data generated will also increase. It will create a huge burden to the current implementation as the scalability is less in this implementation. As the volume of data being transferred back and forth keep on increasing it will become difficult to handle by the existing infrastructure.

Data Security—Transferring data back and forth from the cloud to the edge devices will create a lot of security threats. The path way can be point of attack and intruders can get access to these vulnerable breach points.

Latency—Latency is another issue. As data need to travel a lot of distance to and fro from the cloud devices to the edge devices it will pose a lot of delay in the network. As the requirements of the smart city are very dynamic it cannot afford to have the latency in place. Resource management, complexity, dynamicity are also some of the challenges of this smart city implementation.

To solve these problems a fog computing architecture is introduced in between the cloud layer and the edge device layer. The fog layer provides an extension of the cloud services near to the edge. So most of the computation can be performed by the fog layer itself. Only if there is an absolute necessity data needs to travel to the cloud layer. It will introduce a lot of flexibility to the existing architecture. Since the data can be processed nearby itself no need for data to travel to the cloud servers. So the network bandwidth can be saved. Similarly latency can be reduced to a great extend. This implementation will reduce the security threat as well as data being processed nearby without traveling a long distance. So the possibility of breach points will obliviously reduces. So the overall efficiency of smart city implementation can be increased by introducing a fog layer in between the cloud and the edge device layer. Figure 8 shows a fog-enabled smart city architecture.

As shown in Fig. 8 in a fog enabled smart city has a fog layer placed in between the edge layer and the cloud layer. It will make the computation much easier as the data will be processed near to the architecture thereby reducing the latency and complexity.

6 Facilitating Industry 4.0 Using Fog Computing Architecture

Industry 4.0 indicated the scenario of technology enabled manufacturing. The capabilities of machine learning and artificial intelligence are being applied in the area of manufacturing. Industry 4.0 makes the manufacturing units to be data rich and these data will be analyzed and processed to generate insights which will helps in the business growth. Sensors and iot devices are connected to the equipments for collecting and processing the data. These sensors will transfer a lot of data and handling this data traffic is a challenge. The huge amount of data needs to travel to the cloud servers. This will consume a lot of bandwidth and leads to delay in the network. It also creates a lot of security breaches. A solution to this is the introduction of fog architecture to the industry 4.0 implementation. This fog architecture will be placed in between the cloud and edge devices. This fog layer provides an extension of cloud services near the edge devices. Most of the processing can be performed in the nearby fog layer itself and only for some processing data need to be traveled to the cloud layer. This will result in a lot of savings in terms of bandwidth and reduced latency. IT also

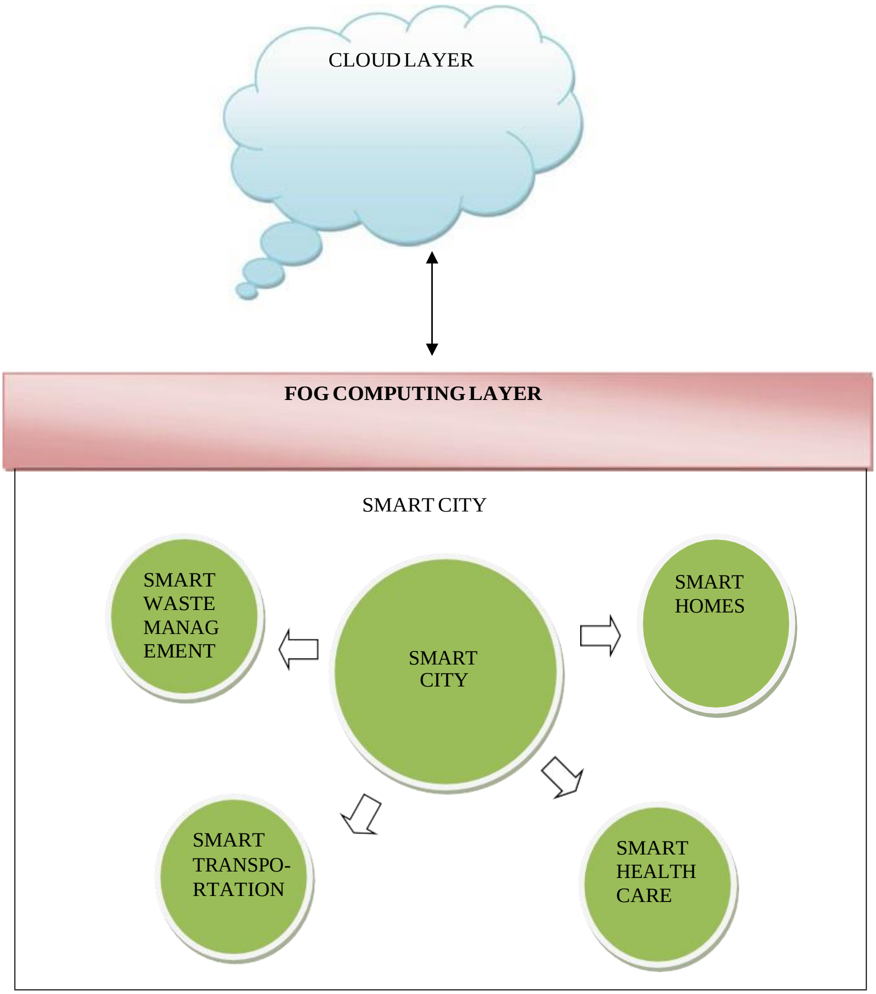


Fig. 8 A fog computing enabled smart city architecture

ensures better security, scalability and heterogeneity. An architecture diagram of fog computing enabled industry 4.0 is given below.

Figure 9 shows the implementation of a fog computing enabled industry 4.0 application. As shown in the figure a fog layer is placed in between the cloud layer and the edge device layer. It will provide better efficiency and security for the industry 4.0 implementation.

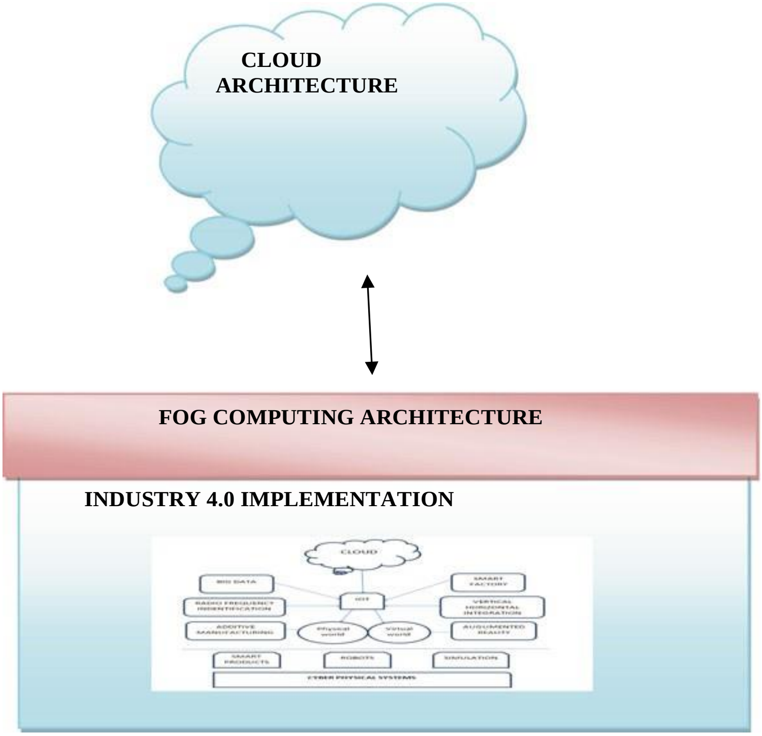


Fig. 9 Fog computing enabled industry 4.0 implementation

7 Three-Layered Fog-Based IoT Architecture for Industry 4.0

This section discusses about three layered fog based iot architecture suitable for both smart city application and industry 4.0 applications [25]. This architecture consist of three layers namely the cloud layer, fog layer and edge layer. This architecture ensures flexibility [26] and scalability. Such a three-layered implementation allows the addition and removal of any number of devices into the edge layer without much impact on the overall architecture [25]. It is very much useful in highly dynamic industry 4.0 and smart city applications. The edge layer consists of all the end devices. These devices are connected with large number of sensors to collect data. These devices together with the sensors are the major components of the edge layer. As the system grows more and more devices will be added to this layer. Scaling is mainly happening in this layer. The next layer is the fog layer. This fog layer provides an extension to the cloud facilities locally. The main components of this fog layer are the iot gateways [25]. Most functionality will be performed in the fog layer itself without transferring the data to the cloud layer. This phenomenon leads to savings in network bandwidth and results in less latency. Only those functionalities which cannot be done at the fog layer will be transferred to the cloud layer for further processing. The third layer is the cloud layer which contains the cloud servers and data centers. If the functionalities cannot be done at the fog end it will be transferred to the cloud layer for further processing. Major computational power of the network is concentrated in this layer. The three-tier architecture in depicted in Fig. 10.

A multilayer fog architecture model is proposed in the paper [27] by authors Jianhua He et al. This implementation is particularly suitable for smart city applications. There are two fog layers in the architecture, one adhoc fog layer and a dedicated fog layer [27]. The architecture is depicted in the diagram given below [27] in Fig. 11.

Fig. 10 The tree tier architecture

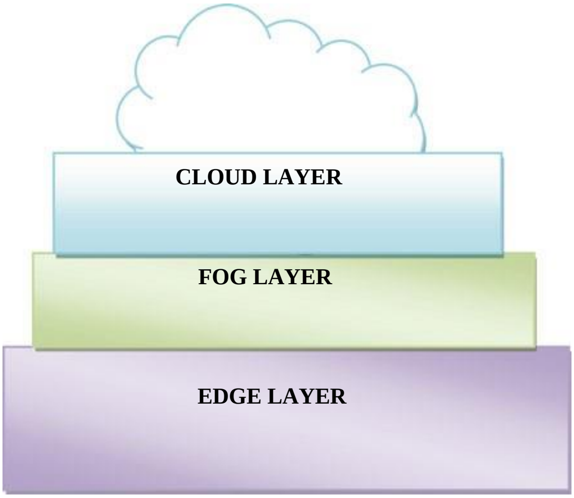
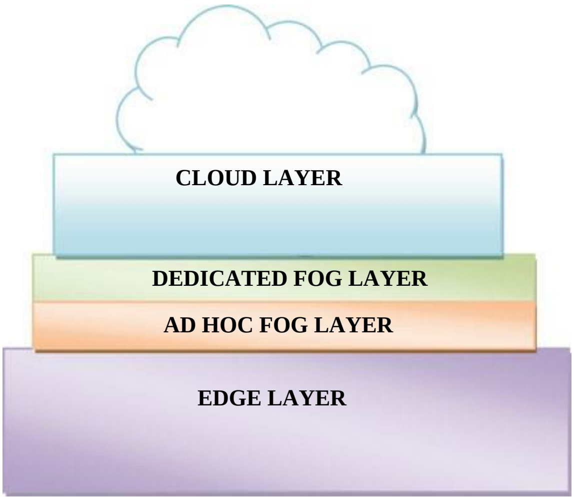


Fig. 11 The multilayer architecture



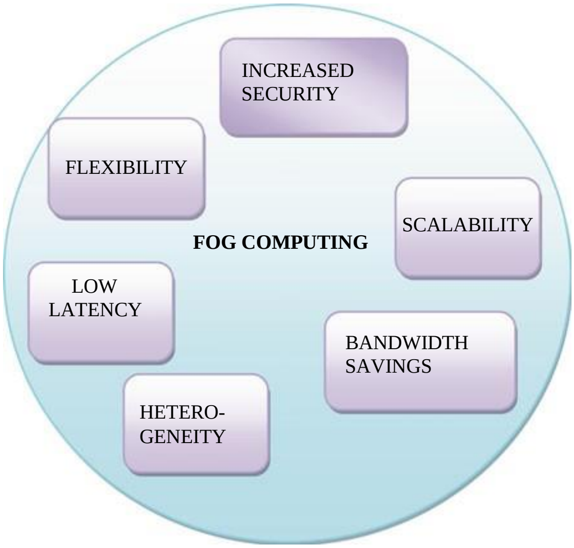


Fig. 12 Advantages of fog computing

8 Advantages of Using Fog Computing for Industry 4.0 and Smart Cities

Since smart cities and industry 4.0 are data-rich scaling implementations, they are relying on the cloud data centers for their needs. Cloud computing has its own benefits in terms of ubiquity, elasticity, unlimited capacity, etc. But transferring very huge amount of data in between cloud and the implementations is a challenging task. It can result in high data traffic in the network. Also causes latencies and security risk. To address these shortcomings fog computing is used. Fog computing brings the capabilities of cloud to the edge of the network. The main advantages of using fog computing are reduction in network traffic, reduced latency and better security. Other benefits are increased flexibility, better scalability and heterogeneity. Both smart city and industry 4.0 applications can be improved by implementing the fog architecture. The benefits of using fog computing are listed in Fig. 12.

Heterogeneity, scalability, security, low latency, flexibility and bandwidth savings are some of the advantages of using the fog architecture.

9 Conclusion

This chapter analyses the applicability of fog computing in industry 4.0 and smart cities. Fog computing architecture helps to reduce the latency of network and also helps in saving the network bandwidth. A huge amount of savings is possible with the introduction of fog layer in both industry 4.0 and smart city implementations.

Low latency, increased security, scalability, flexibility, and heterogeneity are some of the advantages of using this architecture. Proximity to the edge of the network results in decreasing the network latency. Better interaction among heterogeneous services is possible with this implementation. Overall the efficiency of industry 4.0 and smart city applications can be increased by introducing the fog computing architecture.

References

1. Mahmud R, Toosi AN, Ramamohanarao K, Buyya R (2019) Context-aware placement of Industry 4.0 applications in fog computing environments. IEEE Trans Industr Inform 16(11):7004–7013
2. Lin CC, Yang JW (2018) Cost-efficient deployment of fog computing systems at logistics centers in industry 4.0. IEEE Trans Industr Inform 14(10):4603–4611
3. Bouzarkouna I, Sahnoun M, Sghaier N, Baudry D, Gout C (2018) Challenges facing the ındustrial ımplementation of fog computing. In: IEEE 6th ınternational conference on future ınternet of things and cloud (FiCloud), Barcelona, Spain, 2018, pp 341–348
4. Verba N, Chao K, James A, Lewandowski J, Fei X, Tsai C (2017) Graph analysis of fog computing systems for Industry 4.0. In: 2017 IEEE 14th ınternational conference on e-business engineering (ICEBE), Shanghai, China, pp 46–53
5. Fraga-Lamas P et al (2020) Design and empirical validation of a Bluetooth 5 fog computing based industrial CPS architecture for intelligent Industry 4.0 shipyard workshops. IEEE Access 8:45496–45511
6. Hernandes SCL, Pellenz ME, Calsavara A (2018) An architecture of fog computing in smart cities: the middleware E2BS in emergency calls. In: 2018 XLIV Latin American computer conference (CLEI), São Paulo, Brazil, pp 509–518
7. Javadzadeh G, Rahmani AM (2020) Fog computing applications in smart cities: a systematic survey. Wireless Networks 26(2):1433–1457
8. Zhang C (2020) Design and application of fog computing and internet of things service platform for smart city. Futur Gener Comput Syst 112:630–640
9. Singh P, Nayyar A, Kaur A, Ghosh U (2020) Blockchain and fog based architecture for internet of everything in smart cities. Future Internet 12(4):61
10. Zahmatkesh H, Al-Turjman F (2020) Fog computing for sustainable smart cities in the IoT era: caching techniques and enabling technologies-an overview. Sustain Cities Soc 59:102139
11. Badidi E, Mahrez Z, Sabir E (2020) Fog computing for smart cities' big data management and analytics:a review. Future Internet 12(11):190
12. Gia TN, Queralta JP, Westerlund v (2020) Exploiting LoRa, edge, and fog computing for traffic monitoring in smart cities. In: LPWAN Technologies for IoT and M2M applications. Academic Press, pp 347–371
13. Caiza G, Saeteros M, Oñate W, Garcia MV (2020) Fog computing at industrial level, architecture, latency, energy, and security: a review. Heliyon 6(4):e03706
14. Khattak HA, Tehreem K, Almogren A, Ameer Z, Ud Din I, Adnan. M (2020) Dynamic pricing in industrial internet of things: Blockchain application for energy management in smart cities. J. Inform. Secur. Appl. 55:102615
15. Vignesh R, Deepa D, Anitha P, Divya S, Roobini S (2020) Dynamic enforcement of causal consistency for a geo-replicated cloud storage system. Int J Electr Eng Technol 11(3)
16. Deepa D, Vignesh R, Sivasangari SCM, Samhitha BK, Jose J (2020) Visualizing road damage by monitoring system in cloud. Int J Electr Eng Technol 11(4)
17. Samhitha BK, Mana SC, Jose J, Vignesh R, Deepa D (2020) Prediction of lung cancer using convolutional neural network (CNN). Int J 9(3)

18. Nelson A, Neguriţă O (2020) Big data-driven smart cities: Internet of Things devices and environmentally sustainable urban development. Geopol History Int Relat 12(2):37–43
19. Kiwelekar AW, Patil P, Netak LD, Waikar SU (2021) Blockchain-based security services for fog computing. In: Fog/edge computing for security, privacy, and applications. Springer, Cham, pp 271–290
20. Mana SC, Samhitha BK, Jose J, Swaroop MV, Reddy PCK (2019) Traffic violation detection using principal component analysis and viola jones algorithms. Int J Recent Technol Eng 8(3):5549–5555
21. Ramamoorthy V, Divya S, Mana SC, Samhitha BK (2018) Examining and sensing of artificial knee with multi sensors networks. J Adv Res Dyn Control Syst 10(11):115–120
22. Mana SC, Saipriya M, Sangeetha SK (2019) Identification of land document duplication and black money transaction using big data analytics. In: 5th International conference on science technology engineering and mathematics, ICONSTEM 2019, pp 114–118
23. Singh SK, Jeong Y-S, Park JH (2020) A deep learning-based IoT-oriented infrastructure for secure smart city. Sustain Cities Soc 60:102252
24. Malik S, Gupta K (2021) Smart city: a new phase of sustainable development using fog computing and IoT. IOP Conf Ser: Mater Sci Eng 1022(1):012093
25. Peralta G, Iglesias-Urkia M, Barcelo M, Gomez R, Moran A, Bilbao J (2017) Fog computing based efficient IoT scheme for the Industry 4.0. In: 2017 IEEE international workshop of electronics, control, measurement, signals and their application to mechatronics (ECMSM), Donostia-San Sebastian, pp 1–6
26. Barcelo M, Correa A, Llorca J, Tulino AM, Vicario JL, Morell A (2016) IoT-cloud service optimization in next generation smart environments. IEEE J Sel Areas Commun 34(12):4077–4090
27. He J, Wei J, Chen K, Tang Z, Zhou Y, Zhang Y (2018) Multitier fog computing with large-scale IoT data analytics for smart cities. IEEE Internet Things J 5(2):677–686

Fog-Computing: A Novel Approach for Cloud-Based Devices Using Perceptual Cloning Manifestation-PerColNif Taxonomy by Energy Optimization

Rupa Kesavan, Vijayaraja Loganathan, T. Shankar, and J. K. Periasamy

Abstract Computing Paradigms on the Cloud-platform has become an integral service of computing devices that uses internet and big data as the core platform, Computation of big-data under the cloud structure has an emerging need to support the time-sensitive applications over the existing intelligence services. Increase in volume of load leads to increased latency time and reduces the overall efficiency of the services due to access delay. Hence a systematic architecture is proposed for time-line based emergency services to bridge the two platforms namely cloud-server and the edge devices by supporting computational task through fog-cloning over the multiple Cloud Servers and their by ensuring optimal utilization of energy with improved residual sources retained. A novel taxonomy is proposed as a part of minimizing energy consumed through perceptual cloning which aids in identifying time-sensitive data using the location-awareness in the areas not limiting to Military applications, Health Monitoring equipments, Accident Detection Process, etc., thus the latency time of network bandwidth is efficiently reduced with increased resource availability and ensuring improved the quality of service.

Keywords Health care · Accident detection · Time-sensitive data · Real-time applications · Fog-computing · Perceptual Cloudlet Cloning Manifestation Taxonomy [PerColNif] · Energy optimization · IoT-bridging · Arduino-UNO · Fog · bridging hardware

R. Kesavan (✉)
DCSE, Sri Venkateswara College of Engineering, Pennalur, Sriperumbudur, India
e-mail: rupakesavan@svce.ac.in

V. Loganathan
DEEE, Sri Sairam Institute of Technology, West Tambaram, Chennai, India

T. Shankar
DECE, Government College of Engineering—Srirangam, Trichy, India

J. K. Periasamy
DCSE, Sri Sairam Engineering College, West Tambaram, Chennai, India

R. Tiwari et al. (eds.), *Energy Conservation Solutions for Fog-Edge Computing Paradigms*, Lecture Notes on Data Engineering and Communications Technologies 74,
https://doi.org/10.1007/978-981-16-3448-2_6

1 Introduction

Existing cloud services proved it performance and efficiency control in various areas with expensive resources when the volume of data keeps growing every day hence computational time and access time proportionally keeps increasing with respect to storage-process and retrieval. Rapid use of cloud is being increased and is unavoidable in day to day life as technology and complexity in computing the space and time increase in parallel.

Fog computing introduces a novel approach for balancing the task-load and reducing the time of latency over the network connections between application devices and their analytic nodes. In [1] author introduced simulation of Fog-terminal using software, but the proposed model is realized using Arduino-UNO hardware which identifies the Fog-nodes that do not require the bandwidth during data transmissions. Need of IOT and its application is ever-increasing with respect to resource utilization, hence a computing environment with centralized servers is an open challenge [2] for the distributed services. This approach aims at improving the services for real time-sensitive applications as a bridging technique for cloud data sets and Internet of Things (IoT) edge devices. Deployment of this collaborative model can be applied at any industrial areas such as grid-computing, to take an immediate recovery and remedial action.

In Cloud framework, all the data processing is carried out by the central server, which includes additional time, since the data has to be transmitted from the edge-node to central server needs pre-processing of required packets is can be done in the host server, The work of [2] discloses the emerging need of fog-computing as the sustainable technology for promoting smart cities, by resolving challenges listed on cloud-platform. It is highly difficult to rush terabytes of data from the edge-node to the server cloud and back. Hence to yield better results of these drawbacks, an extension of cloud paradigm of computing called fog mechanism, is promoted. In the proposed work, Tri-Layer Architecture as shown in Fig. 1, shows inclusion of Core-layer as several cloudlets preferably Health care data center and Perceptual Layer

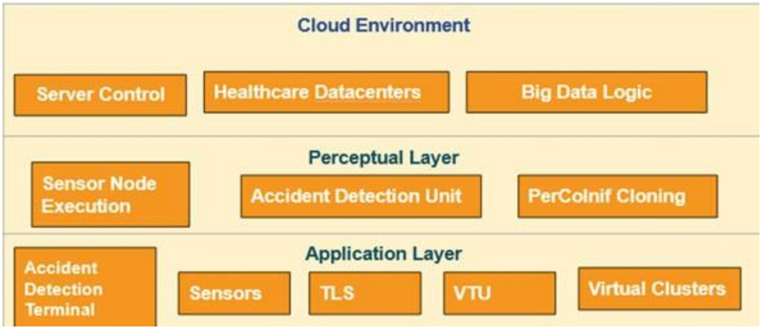


Fig. 1 Tri-layer bridging architecture

involving the data processing of instances done in the node, if higher computing power for the data is not required it is done partially and if the data requires high computing power data is moved to the central server for the remaining computations. This generally lowers the time engaged with the process and is more effective as the central server is not overloaded.

In Healthcare IoT applications have given an organized methodology towards improving our medical care administration [3, 20]. This is accomplished by conveying pervasive investigation and to update the information on the haze gadgets progressively prior to sending the required instances to the cloud for further investigation and decision. Fog computing acts as a intelligent bridge to give progressed strategies and administrations, for example, appropriated capacity and interfaced data mining. In [3] the detailed analysis on electrocardiogram contours extraction that assumes a fundamental part on processing the instances based on chronic illness in heart is disclosed by the author. The outcome recommended that implementing fog computing accomplishes relatively reduced latency and immediate response at the nodal-devices connected to the network with more than 90% bandwidth efficiency. To encourage simple admittance to medical care administration for the affected patients, a customizable sensor equipped with the patient‘s body connecting to the fog-edge was proposed [3]. The fog as the bridging network is utilized to upgrade the framework by facilitating various administrative tasks not limiting to Health care, Automation, and detection of critical instances on the Military field, the evolving drone applications can be interfaced to the fog-paradigms [19]. This assistance also adds the incidence location for easy trace. Perceptual framework for immediate dataset with minimal bandwidth at edge-devices is proposed by deploying Fog computing as an interface on the Arduino hardware between virtual devices and the server-cloud, this approach will limit the downtime and the migration time to ensure resource and service availability to the end-users. Accordingly, in [4] author discuss about smart pre-copy live movement approach is introduced for VM relocation, which assesses the personal time after every cycle to decide if to continue to the stop and-copy stage during a framework disappointment or an assault on a fog computing node. In [4], technique to analyze and send the frequent video applications and services, ranging from proxy-assisted rate adaptation for caching the smart features are proposed and ensures with the improved performance using the fog-mechanisms, but still in [4] algorithm for frequently accessed data set is fogged up, rather in our proposed work we target with the time-sensitive services to recover the critical spots using the fogging feature based on two customizable modes of sensors, that guarantees much better results with optimal latency time and best quality of services.

1.1 Fog as Eminent Computing Paradigm

The growing technology Cloud is a framework that enables the rapid services for increasing volume of real-time data volume more than Terabytes, still the cache mechanism of recently accessed nodes involves complexity with computation of preserving the resources, so resource management is a challenging part for the cloud paradigm with increased resources [2], thus for the virtual devices on the cloud platform require a better approach to provide geographical computation with available bandwidth. Although these complexities is addressed with the MQTT services [1] for resolving the lost of packed on the TCP-switching including the switches as the broker nodes on the local server using the fog-mechanism, still this incur the increase in cost as well as better latency cannot be achieved. Hence to overcome these issues a novel approach for the wireless access points are targeted as the edge devices, out proposed work appreciates variety of edge devices not limiting to the sensors but also the android devices, mobile phones, gadgets supporting the wireless access can be the edge device on this edge-computing paradigm.

In [5] a 4-layered architecture on fog-computing is established to transmit sensor data in an optimal network, the prototype is developed on a pipeline system and applied Gaussian Model to compute the task on fog-nodes, still the model does not support dynamic node features and location awareness. In the proposed work new paradigm collates the computational task on the cloud server for the timeline-based applications to overcome these limitations, GPS sensors Fig. 2, are equipped to realize the location awareness on the fog-nodes that yields better performance for the improved load, also detection of intrusions on the network channel can be resolved using the dynamic node interfacing with the similar nodes available on the virtual cluster nodes. Thus virtualization framework is introduced to reduce the downtime.

Fog architecture gives numerous advantages to the applications computed on the edge devices by effectively offloading the task on the minimal bandwidth for the Optimal resources on the edge node, thus fog-benefits the load balancing, supporting the operations on the high-end cloud server-bridging the edge devices as mentioned in [6]. Since the energy requires to select, organize and process the partial data from

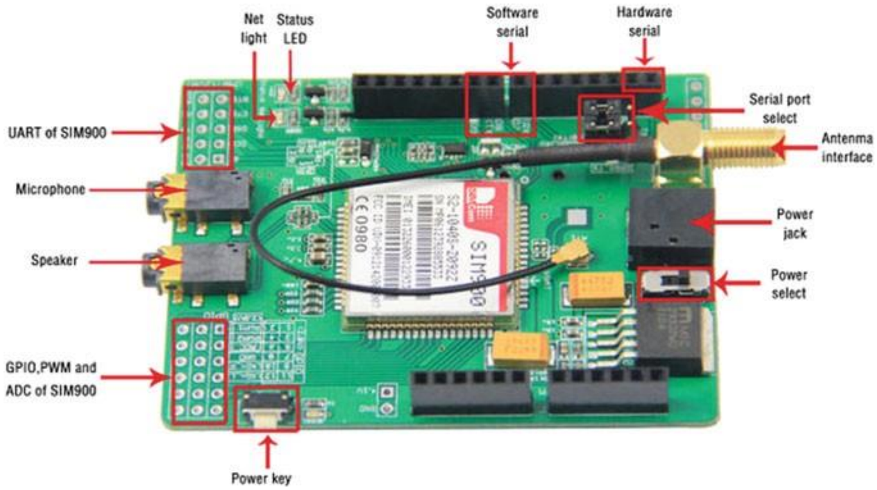


Fig. 2 GSM–sensor specifications

the heavy storage is operated for the part of the ideal target data, the computation task ensures minimal energy for the access points on the network, thus assuring less latency time, as the computing time for the multiple resources on the network.

The paradigm of fog computing to put-forward in emerging services yields improved performance with the volume of fog-nodes increased over the computing paradigm. The increased availability of fog and cloud assets is a pre-requisite factor for the defect caused during the load balancing. In the proposed taxonomy cloning process is attained through perfect sensor nodes operating at frequency of copying data sets from the source server $Point_A$ based on timeline, thus improving the execution speed to attend the victim with reduced communication delay their by saving the lives earlier by serving on-time.

A three-layer model is proposed as mentioned in Fig. 1 which consist of layers such as core-layer is the cloud computing paradigm that includes multiple servers, they can be any private or public clouds servicing as Saas-Iaas clouds, these service providers enable various high computing paradigm for data monitoring, data managing, data analytics on more volumes namely Mega-bytes to Tera-bytes, in our proposed system the core-layer includes Health Care services as the data center to support the services interfaced with the Traffic-light System equipped with the Accident Detection unit on the Arduino board.

The Perceptual process of the both the algorithms works on creating location awareness through the GPS sensor equipped at the Traffic Light System embedded with the Accident Detection Terminal.

The cloudlet server at the source access point determines the location of the vehicle colloid incident as the data set stored in the cloud captured from the Accident Detection Terminal [ADT] through the sensor node equipped in the Traffic Light System and thus the fog node ensures raising alert to the Hospital data center to take remedial action for reaching the ambulance/recovery team on time. This mechanism is facilitated to generate the alert to the Hospital server connected to the nearby localities to recover the spot without the communication delay thus the introduction of novel taxonomy ensures reaching the location through location awareness feature of the fog on-time, and facilitating the remote server to take necessary actions in reaching the hospital in time to save the lives.

Middle layer is the platform where the proposed taxonomy is applied on the fog edge, Perceptual layer is integrated with the cloudlets peripheral node with the timeline dataset and the interface the Health Data center, this layer work on the virtualized framework.

The bottom layer is the Virtual cluster with the interconnected component varying wireless access points, connecting the sensors with the Arduino hardware.

2 Related Works

2.1 Cloud Computing Era in Health Care Services

In view of the Utilization of the existing technology, cloud stands as one of the leading computing technology interfacing several application devices locally and remotely. In [2] the emerging need of the post cloud mechanism is analyzed with the detailed examination of performance efficiency and the load balancing for the heavy computational task on the in-operable bandwidth, these restrictions of the cloud leads to introducing a promising technique for heavy computational operation for the real-time data, thus infrastructure that co-operating the edge device with the cloud server is given as insight by the author [2], similarly, in our proposed framework a novel taxonomy is introduced with the aim of bridging the real-world sensor-based device computation with the high end cloud server to share the workload based on the time-sensitive application. The author discuss about the energy efficiency computation and brings the importance of reducing the latency time with the efficient resource management as the present demand for cloud platform [2], the supporting feature of fog on the cloud as the peripheral function will address the challenges of the cloud as the leading edge of future techniques, in our proposed work the fog is equipped with the customizable sensor modes to detect the ideal dataset from the multiple servers.

Considering this we proposed our work on fog computing with healthcare application to track the computing tasks in healthcare, processing time, and security factors. In [5] author introduces an three-layered architecture for the fogging the sensor node to stream line the pipelining data using the optimal communication services, here the packet delivery is limited for the device coverage ranging up to 50 km over the cloud storage, hence we propose the services through sensor nodes to switch over the multiple cloud servers, also to interconnect the neighboring nodes during downtime.

Also, the fogging role for the health care services becomes an integral service for the patient monitoring systems with a layered architecture as a part of promoting smart cities. Cloud facilitates the services through the local monitoring and reporting the dataset to the nodes, devices embedded with the current system.

In [3] fog-based services are discussed for assuring efficient performance on monitoring the patient‘s Heart rate for definite patients to attend the chronic diagnosis, and this is been simulated using the tool to analyze the performance. Whereas in our proposed work, the observance is made using customizable nodes working on two different modes for reporting the remote server, and the functionality of the sensor is simulated on the Arduino board.

2.2 Cloud Computing for Vehicle Tracking–Accident Detection Systems

In [6] smart architecture is introduced by the author to track the vehicles using the sensors equipped in Traffic Light system using the GPS sensor as shown in Fig. 2, and these existing application is deployed on the cloud structure has vulnerability to the DOS attacks at higher data rates, thus addressing this issue in our proposed taxonomy through the location awareness of the fog-based interface with the cloud server through the GSM node sensing the incoming incident on the fog node, below is the descriptive peripherals of the GPS sensor.

The integration of IoT technologies in the transportation domain has facilitated the lifestyle of drivers. Cloud computing that gives numerous advantages to IoT devices, including high-performance computing, storage infrastructure, processing and analysis of IoT data in real-time based on the current context of IoT sensors resulting in robust, smart, and self-configuring.

In [6] importance of Migration towards fog is discussed, authors insists of the necessary service for the edge computing paradigm, that encourage an ideal service for the cloud and fog resources. Existing work deals with FSDN which is a Vehicular based Ad-hoc Network (VANET) relying the fog features to collaborate the attributes such as connectivity, mobility, scalability, and flexibility of VANETs. But the mechanisms as optimization of their source utilization, service hosting, migration, and replication are not discussed [13]. To make these mechanisms more meaningful, it can be resolved by expanded resources utility, reduce latency and network traffic by exploiting the advantages of fog computing.

In the related work [6], fog-paradigm is discussed for introducing a system architecture for Vehicular tracking over the cloud-framework, and the author analyzed the intrusion attack, vulnerability on the cloud nodes, the proposed architecture provides better energy efficiency due to fogging of server instances. Day to day number of vehicles keep increasing an emergency mechanism with automating and controlling of traffic is required for sustainable development. Thus in [7] author introduces smart traffic control considering road side unit as the fog terminal on the virtual cluster, here the system proposes detecting pot-hole on the road side unit, capturing these instances for preventing the damage for the vehicles.

In [7] author Diffie–Hellman technique is introduced to interface the mobile sensors with the Road side unit for processing the input instance at optimal cost for encrypting and decrypting the captured instances through location-based encryption. Also this work includes encryption namely signcryption process involving certificateless aggregate mechanism. Whereas in our proposed work we introduce vibration sensor, GPS sensor interfaced on the Arduino UNO board, and the flow of work is depicted in Fig. 3, which performs detection of colloid incident on the ADT unit, the Perceptual Cloning process is performed on the edge-device to ensure reduced Latency time.

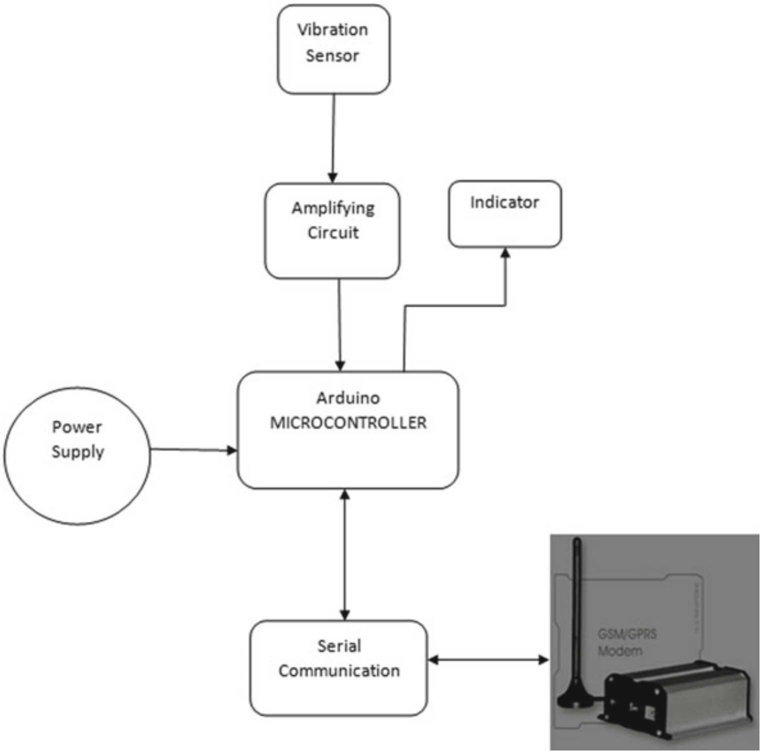


Fig. 3 Data flow diagram of hardware architecture

2.3 *Fog Computing Versus Cloud Computing for Time-Sensitive Applications*

Technology today means computing big data on the Cloud storage ranging from simple small gadgets to any real-time server computation relies on the cloud computing paradigm, beginning the storage, manage and processing involves data collecting, computing, and analyzing everything done on the single-point cloud. Information from the end clients are put away in cloud by means of sharing the public / private computing space on the cloud that can be made available at anyplace for the end user to access. Still, the load balancing is being a challenging task [2, 17], here we expand the idea for supporting cloud computing task that provides a promising approach to increase the bandwidth by reducing the latency time. Proposed idea administrates the incident on the edge of the virtual cluster to diminish interlude and traffic existing on the network environment.

A similar cache of cloudlets task serves as the interconnect service for the existing computing mechanisms that distinguish by providing optimal latency with location awareness at a wide spread of edges facilitating mobility and real-time strategies.

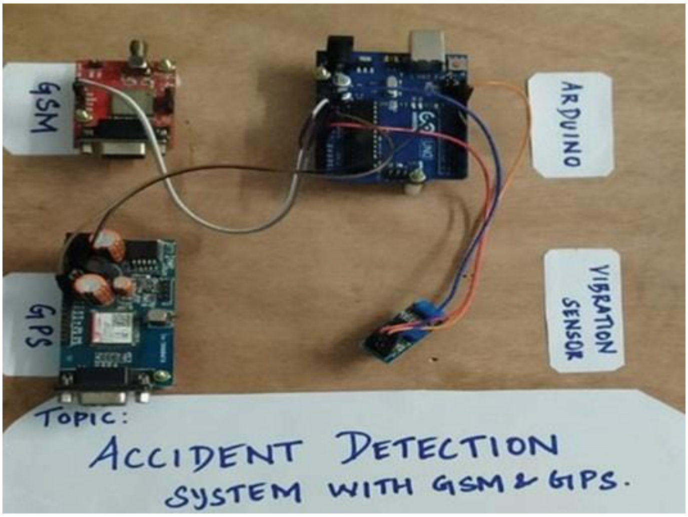


Fig. 4 Hardware architecture

In [6] author performs various analysis of parameter by review done for more security and privacy issues ensuring the challenges on future research for fog based services, here the author introduces restrictions with energy efficiency for dynamic node transmissions, resource management for the terabytes of data reaching the multiple cloud servers, hence we propose a novel approach to address these requirements using load-balancing technique [8] for fog-node to node by introducing dynamic node communication in the virtual clusters with a remarkable quality-of-service (QoS), through edge routers with wireless access points. In [2], the author deployed unique surveillance systems to put-forward the observance on the fog devices in the current devices stored earlier to transmitting the information at the cloud end for further analysis and diagnosis.

A smart gateway is provided with advanced process on the open-source interfacing into hardware input for data analysis, the interfacing connections are present in Fig. 5, leading to low interruption and real-time response at the edge of the network with more than 90% bandwidth efficacy. In the habit of facilitating quick access to remote data center for the patients, a body sensor network in fog computing was proposed in [2]. The upgraded technique in fog computing is used to offer distinctive services not limiting to Health care services, vehicular tracking, monitoring patients heart rate, pressure at emergency units, feature extraction, over open-source database and graphical interface for assuring generated observances in a visualizer to diagnose at the instant that improves the quality of services.

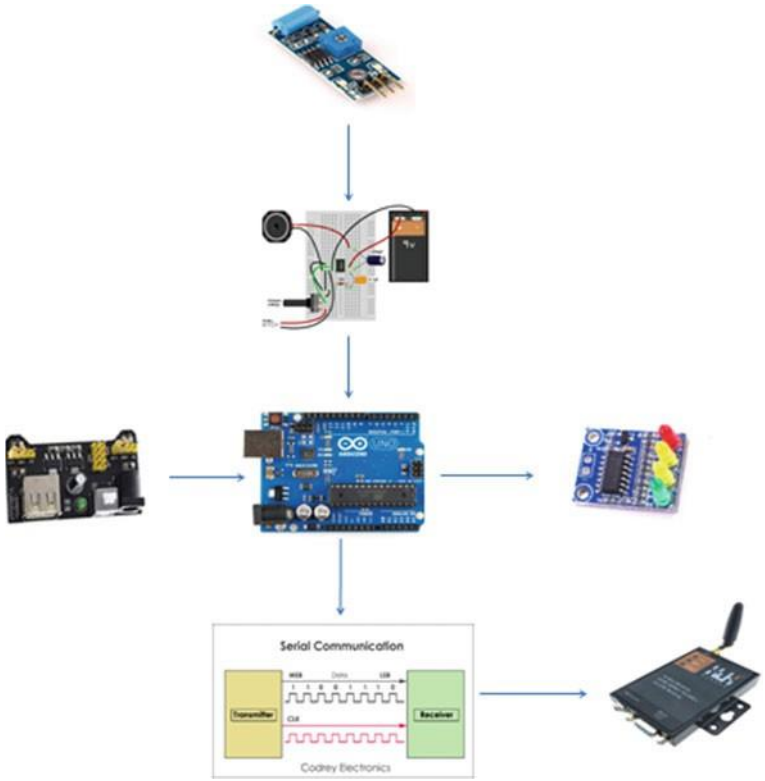


Fig. 5 Sensor–Arduino-UNO inter-connection flow

2.4 *Virtual Clusters—Fog Bridging Cloud with Edge Devices*

Our work proposed the conception of fog computing and the fog node consignment according to various stipulations such as heterogeneity, QoS management, scalability, and progression to attain location awareness, low suspension and energy optimization.

Fog computing uses intermediate layer between devices and cloud storage. In our work, we use fog nodes to save the bandwidth and reduce latency for content delivery to improve the quality of health and accident detection services. Fog computing is recommended to manage the IoT resources and allow cloud platforms to act according to the current situations of end-users, with the help of fog nodes to schedule competently, stimulate QoS, allocation, and managing of cloud resource, valuing, and security. In [2], evaluates the cloud resources necessary for processing requests according to the behavior and historical record of users, this leads to Minimized cloud resources underutilization. We provide an systematic mechanism for pre-processing the data before sending to the cloud by storing it in the fog nodes, leading to mobility and invulnerability. In case of healthcare monitoring system [18], we authorize patients to access health care services at competitive costs resulting from a faster and more structured health care feedback to patients which results in high security and privacy of patient's data. Comparably, FSDN is a Vehicular Ad-hoc Network (VANET) based on fog features for supporting connectivity, agility, adaptability, and flexibility of VANETs [2].

We resolve challenges in VANETs by providing better connectivity, adjustability, pliability, and Intelligence through fog node interrelation. Fog to fog interaction also slows communicate with each other, and it reduces the overall end-to-end delay. Communication and synchronization between fogs or between the cloud and the fog layer, catching mechanism, selection of services within F2C, Investigation of other edge technologies.

3 Tri-Layered Proposed Architecture

Proposed Architecture is a three-layered architecture that is configured in to two base layers core-layer and virtualization-framework as depicted in Fig. 6. Foremost top layer is the set of multiple cloud servers, here Hospital Data Center, and remote

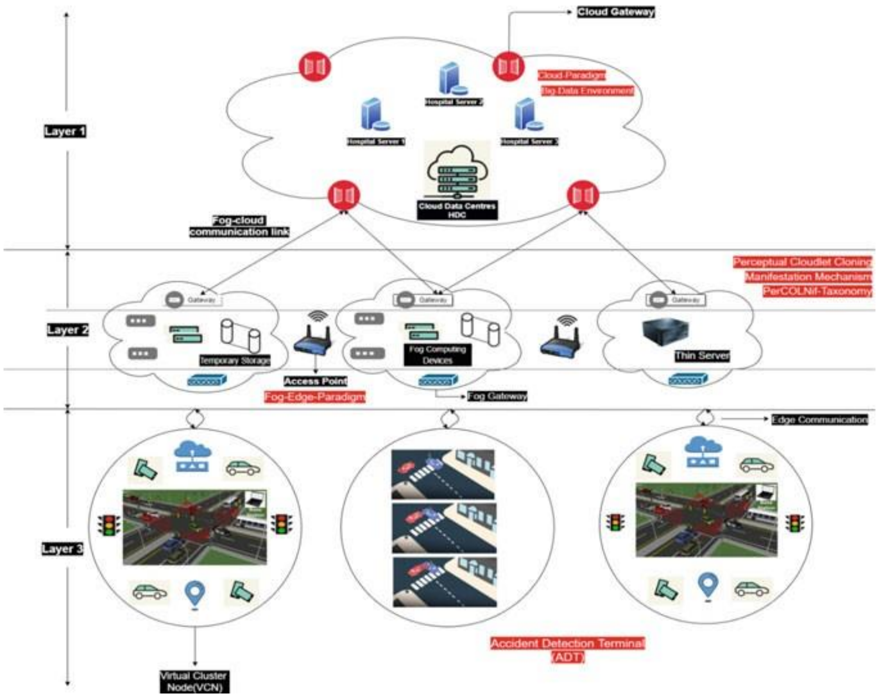


Fig. 6 Working of PerColNif-taxonomy-in–three-layered fog-based architecture

server is interfaced. The second layer uses Perceptual Cloudlet Cloning Manifestation mechanism-PerColNif Taxonomy which includes Gateway, Access point temporary storage, fog computing device, and fog gateway to communicate with Layer 3-Accident Detection Terminal. The third layer consists of end-users, which is made as virtual cluster node (VCN) to communicate with the fog devices. Making the servers available directly to VCN, simple access of data and retrieval is easy leading to low consumption of energy and all the network equipment [3]. Author has also developed a MILP model to optimize serving the health monitoring application patients from these two layers, so the total energy consumption is minimized [1]. In this paper, author proposed a new algorithm named Software-Defined Networks (SDNs) enabling programmability to the network switches, and a centralized control-plane controller to enable routing decisions by appropriately utilizing available network resources. The SDN controller can serve as a platform for performing analytics by parsing the MQTT payload contents to retrieve topic and associated data value transmitted on the optical pipeline structure, this mode of streaming the data packet leads to packet loss, as well as the increase in latency time. In our system we proposed a new mechanism called Perceptual Cloudlet Cloning, helping us to communicate from VCN to the Cloud storage which is a big data environment.

The outcome of consumed energy is computed in our model with respect to the time-line based target value as mentioned in algorithms, proposed algorithm also computes the delay time, residual energy of the node inside the virtual cluster. In [9] author describes new Fog-layered architecture called n-type-model, in which Fog nodes are gathered based on the locality and edge device functionality included inside the fog environment. In our paper three-tier architecture is used, the bottom layer of the system architecture edge devices that are stored on the virtual clusters, these may be smart devices, mobile devices, traffic light systems where the GPS sensor is equipped with to collect the colloid incident.

The middle layer of the proposed architecture involves the Perceptual data collection working on customizable two mode of sensors, the relax-mode, and quick-mode, the devices that support in this perceptual layer are thin devices, that efficiently computes the task at minimal bandwidth, consuming optimal energy thus transmitting necessary results to the remote server, their by achieving the task more efficiently in a smarter way.

4 Proposed Taxonomy—Fog-Computing–PerColNif-Taxonomy

The Cloudlet Cloning taxonomy provides two basic conditions with which fog node is positioned, to run the cloning process for the cloud data based on the timeline value [16], for our work we set the timeline as the recent time for 2 days of data sets, hence the latency for data storage, processing and retrieval is comparatively less than the cloud center, thus ensures a low downtime when the dataset is cloned for contemporary use.

The data rate for the manifestation of the cloned copy is relatively higher for the increased workload in the Relax-mode of sensor nodes executed till next manual reset of the timer, but still pertains data captured with the immediate effect.

For the maximal efficiency of the data rate, ideal value is set to analyze data set till recent captures for only last 24 h, hence cloning process under the quick-mode captures the dataset in the cloudlet based on this condition prevailing higher bandwidth with optimal energy consumption, leaving nil residue energy, also the communication delay with this node executed aims reduced delay related to other paradigms. The cutting-edge of target set with the timeline-based perception will prove the necessity in fog mechanism in the health care services bridging with the Accident Detection/Tracking process.

The ideal value with the condition set to perform the clone from the ADT that transmits the necessary data set to the Hospital Data Center using Timeline-Based services ensures high availability, with respective to repetitive perceptual manifestation process by estimating the downtime after each circle of time the base-line value is attained.

Point_A Sensor node fixed at Traffic Light System [TLS] is the Source Cloudlet server that acts as access point at the computing edge, where Point_B is the Hospital Data Center/Hospital Cloud server which is the recovery unit, Accident Detection Terminal [ADT] is the fog unit that interfaces the transmitted data from the TLS to the HDC through PerColNif Taxonomy working in two superior modes namely Relax mode and Quick Mode based on the Timeline correspondingly The Timeline-based perception from the cloudlet helps in evaluating the cloning features to succeed with the fog-node recurrently analyzing dataset on the Source cloud Point_A and destination cloud Point_B successively, thus through the encircled conditional dataset manifestation process can attain efficient threshold to generate higher data rate in an optimal downtime.

It is observed from the several related works that future era fog and the existing cloud mechanisms drive the virtualization framework for computational tasks. Our proposed foresaid our taxonomy PerColNif methods are designed to attain optimal efficiency with respect cloning time and downtime without recourse to their various ideal dataset volumes. We believe that predefine a Timeline-based Cloning process for the application-specific emerging need of datasets relying low-cost, reduced latency time, reduced residual energy will significantly provide an efficient performance under the fog-computing platform. Hence this novel approach implemented through two modes based on the size of the cloudlet predictable cloning operation can be deployed to estimate the energy efficiency of the fog-node with the access point and the downtime during to node–node computation.

4.1 Perceptual Cloudlet Cloning Manifestation Process [PerColNif]

Growth of computing paradigm is tremendous in various applications of big data and still several challenges arise for resolving existing services based on timely need varying with the real-life scenarios. With detailed study of several computing paradigms across different applications, services made by several authors in recent years, we propose a framework for one of the major application bridging the Health care services with the traffic light system. The proposed framework focuses on bringing attention over immediate recovery on the Accident location by integrating the Traffic Light System as Accident Detection Terminal [ADT] with sensor-equipped their by interfaced with the Health care centre – Hospital Data Center [HDC].

The existing scenarios [10] work with the Big Data-Cloud Computing paradigm, whereas we felt special attention can be given for providing rapid services to save lives, reducing the communication delay, to attend immediately the victim who met with the accident, requiring instant hospitality without delay in communicating remote server for immediate decision. Thumb-Rule to accomplish accessibility is to introduce the Fog processing in geographical hubs that rely upon cloud workers, stockpiling, and organization. Contingent upon the limit of the cloudlets, over-provisioning may be restricted and can directly affect the expense and the presentation of other conveyed client applications.

Here we introduce the cache process by a novel approach-defining a new taxonomy over any cloudlet platform-like private or public cloud supporting Services-as-a-source, Enterprise-as-a-source environment. The cache processing mechanism is facilitated by “Perceptual Cloudlet Cloning Manifestation [PerColNif]” process in which the fog-nodes rely on the timely recovery of surveillance from TLS [ADT] and The reason to choose the perceptual cloning technique for the selected work is that, existing [10] work of Health care services implemented using big data-cloud computing has constraints with increased latency time for being stored big volume of data, expensive data unit as single point of storage leads to high bandwidth, leading to security issues. Hence we decided to resolve this by proposing a novel taxonomy with selecting, organizing, interpreting [manifest] and disseminate the necessary immediate data that is been stored within timeline.

To achieve this process, the sensor nodes are configured to work on two customizable modes, the primary goal of the chosen application is focused on ADT-Accident Detection Terminal, to ensure persisting the detected recent incidents to made available from the cloudlet as the clone-copy. Rather retain the frequently accessed data alike the cloud paradigm.

4.2 PerColNif-Algorithm Implementation

Thus the Perceptual Cloudlet Cloning Manifestation [PerColNif]" process is configured to process with two operating modes:

1. Relax Mode—To perform the Clone process for the maximal node capacity that executes the edge computing to run the sensor node with reset timer till next alert.
2. Quick Mode—To perform the Clone Process for the timeline prevailing the contemporary data and executes the sensor node on the edge device as default mode. For the best energy efficiency, the Cold-mode of the sensor node can best fit for the timely based applications relying instant fog services.

[PerColNif] taxonomy can bring an emerging need for future research on various deadlines such as raising alarm call for Patient Monitoring [3] systems, Recovery units during accidents on Highways [6], call for Patrol services, to perform the data analytics based on the present data.

4.3 Replacement Services–Node–Node Communication

In this novel approach fog-node is the bridge between the two active cloudlets Accident Detection Terminal [ADT] and the Hospital Data Center [HDC], which works on timeline under two modes, hence improved efficiency with the data rate and ensuring the resource availability for the access point on the virtual clusters, the existing systems [3, 11] works on automating the services proved as simulation work, where our proposed architecture is an hardware worked on Arduino, and a novel algorithm [PerColNif] is proposed to prove the energy efficiency at optimal latency time, reduced downtime comparatively.

Whenever the efficiency of the fog-node performance is observed to reduce, a test condition is generated to identify the nearby fog-node that replaces the default fog-node for temporarily bridge the ADT with the new hospital server. Meanwhile whenever a fog-node behaves like the acting fog-edge, the minimal efficiency of the actual fog is reported to the primary cloud for rectifying the down-node.

5 Fog-Computing Implementation

5.1 Algorithm: ADT-Relax Mode-Quick Mode Sensor Node-Accident Detection Unit

```
Algorithm:  Point_A  -Edge  device  equipped  with  Sensor  ADT-
execution
Source Node_id: sensor node_id;
Input_data_value: sensor node_id "colloid datasets"
Er: total ADT_dataset_energy in joules
Destination_node: Remark Data Timeline left

1: t1 initial =0: Sleep timer_mode
2: while  Er> Et1
   do
        Source_Edge_Sensor has still not reach maximal energy,
        ready to Select, Collect, and Disseminate.
3: m_recv():
        Sensor gathers dataset from cloudlet of source Point_B,
        dataset_alert_message  from  Destination  sensor  for
        relay.
4: if Node_ id == Found "Sleep"
         then  Source_node_Sensor in  marked sleep_mode
5: wakeup()
6:  Start()  //Read  Consume  dataset  energy  for  waking  up;
         Quit_sleep
7: end if
8: if Rm Is  input dataset then
9: send(data) //Broadcast own data
10: collect() // Read Consume dataset energy on activity;
      11: else
12:  ignore()
      13: else
14:  disseminate(Rm)//Acknowledge  the  message  received  on
relay
15: collect() -Consume dataset energy on activity; Eactive
16: end if
17: end if
18: t10 ; sleep() // Reset dataset sleep timer
```

5.2 Algorithm—Fog Node Execution

```
Algorithm: Computational task Executed at Fog node
Input data ft: fog node; b: local Edge device
Output : Transmit(dataset)
1:t2 _   0
2: while true do
3: if ft  In State Wait then
4: t2Exec(Recv()). Receive and execute job
5: Send(o, b). Send outcome to loc_ Fog node
6: end if
7: if _ max(ds)  then
8: t2_   0 ; Send(mh,b). Reset timer and send Curr_dataset to
loc_Fog_node
9: end if
10: end while
```

6 Advantages of PerColNif Using Fog-Computation

Fog-Computing paradigm is the evolving edge which serves as the concrete between Cloud providers, edge device-logics and IOT platforms, which is the significant feature of fog although several advantages stay behind, and perceptual cloning manifestation based on the stipulated timeline operating in two modes resolves the security and privacy issues by consuming peculiar resources from the cloudlets.

After the detailed study of related works [4, 9, 12] it is observed that privacy and data Security is a challenge to address in the cloud, thus we refer to the advantages of the future Fog-technology to ensure the Optimal Latency, facilitating Location awareness-based feature extraction, location-based processing encryption, assured authentication, optimal energy efficiency, effective residual energy, providing rapid service for time-sensitive analysis, privacy issues of the cloud-based framework are resolved using location awareness using edge computing (Fig. 7). From the analysis of [11], author discloses beneficiary services p of the fog-computing in a secure Intelligent Traffic light control systems revealing security, device friendliness, privacy protection, and reduced latency.

Thus, our proposed system ensures the below goal attained for the perceptual cloning of cloudlet dataset based on the timeline attribute operating on Relax-Mode and Quick-Mode.

Security: Related work on Health care and Vehicle tracking exist individually but proposed fog-paradigm PerColNif ensures secured data availability with location-awareness, since nodes are customizable on Relax mode and Quick Mode.

Privacy protection: Node–Node communication through the location-based encryption scheme promises privacy protection in the virtual clusters.

Low Latency: The Open-N-Fog architecture [6] ensures data processing and manifestation proving optimal latency. Therefore, fog computing efficiently attains the demand for services ensuring instant-data processing for timely-data, promisingly that need high attention for time-line based-sensitive applications.

Energy–Efficiency: PerColNif-perceptual cloning mechanisms allow manual reset timer in the Relax-mode, and default data set cloning performed in Quick-mode, thus ensures agile cloning performed with reduced communication delay leaving optimal residual energy, and proves Optimal energy utilized in accessing the cloud-data.

7 Energy Efficiency Performance Analysis

Fog computing has emerging fascination in accompanying the growth of technology to substantiate the real-time-applications on the cloud-platform, Optimization of energy in the close fog node shall be eminently optimized by perceptually cloning the data sets from the ADT. Still the left out bandwidth for the neighboring nodes demands the request for sharing the load which incurs surplus energy and communication latency is incurred due to the data communication between the edge-node and the cloudlet (Fig. 7).

The proposed work facilitates time-sensitive services to operate on two-mode which ensures agile task computation at higher bandwidth, thus reducing the overall energy consumption and latency. PerColNif-Taxonomy considers multiple fog devices for various time-sensitive emerging applications, to load the workload for the colloid incident on the vehicular tracking–Traffic-Light System. Related works [10] perform smart pre-copy for dirty pages that leads to increase the task-loading for specific task would affect the performance of few related applications. To resolve

Privacy Issues Addressed	Cloud Platform	VANET Framework	Networking Services	FOG/Edge Computing	Virtualization Platform	IOT Platform
MIM	X	X	X	✓	X	○
DOS	○	○	○	-	○	X
Fault Tolerance	○	X	○	✓	✓	X
Location Based Awareness	X	X		✓	X	○
Vehicular Network	X	○	○ ○	✓	✓	✓
Healthcare Services	○	X	○	✓	✓	✓
Agriculture Farming	○	X	X	✓	✓	✓
Node-Node Computing	X	-	✓	○	X	○
Resource Availability	○	X	X	✓	○	○
Resource Sharing Allocation	○	○	✓	✓	○	✓
Location Based Encryption	○	○	X	✓	X	X
Residual Energy	X	X	X	✓	X	○
Identity Privacy	X	X	X	✓	X	X

Key-note: ✓ services strongly available, ○ partially supports the services , X – not available

Fig. 7 Edge devices supporting services metrics analyzed with varying processing platforms

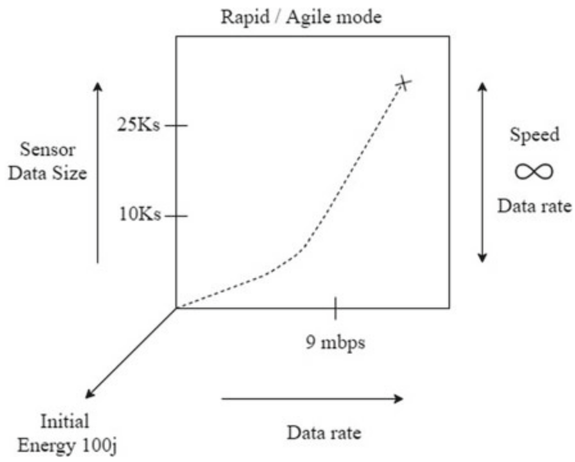


Fig. 8 Increase in packet size on edge computing Increase in speed

this issue, ideal condition is set to fit the fog-node capturing colloid-incidents by estimating energy consumption, and performs cloning decision based on perceptual selection of dataset. All cloning tasks when collected they are predefined with the ideal value and with a dual-phase condition to dynamically facilitating the updates on the platform to execute in the virtualization framework under permissible bandwidth and on the timeline-based delays caused by varying devices with multiple tasks.

Fog-Computing facilitates location-awareness services for attaining time-sensitive operations that play major role in Health Care services meeting the ratio of data rate with the perceptual selection of specified data set on time-line.

The proposed work is simulated on the Arduino hardware (Fig. 4) and the performance of our proposed taxonomy can be simulated with different bandwidth availability (Fig. 7). Each cloning process involves selection of dataset meeting the deadline, and mapping the colloid incidence task set consisting 100% utilization. The vertical axes, respectively, indicate the meet ratio in Fig. 8, which is deployed on various network-enabled platforms.

In Fig. 9, whereas the horizontal axis depicts the various normalized bandwidths. In this experiment, the network bandwidth between the cloud server and local devices is set at 12 MB/s (100% bandwidth), 9 MB/s (75% bandwidth), and 6 MB/s (50% bandwidth).

Every access points on the wireless sensor modes at the increased bandwidth execute on two modes with require low latency for the transmission. Computation of task done within the local node of fog-device but local node with the maximal energy consumption as presented in optimal energy consumption for maintaining the device life. It is observed that Cloning factor as ideal target value at 0.8, the number of incidence collided as tasks in Quick-mode of sensor promises the optimal energy consumed, but the meet ratio decreases slightly by about 1%. Packet delivery ratio shows the improved performance for varying network platforms with implementing the novel cloning algorithm ensuring low latency and downtime may be achieved.

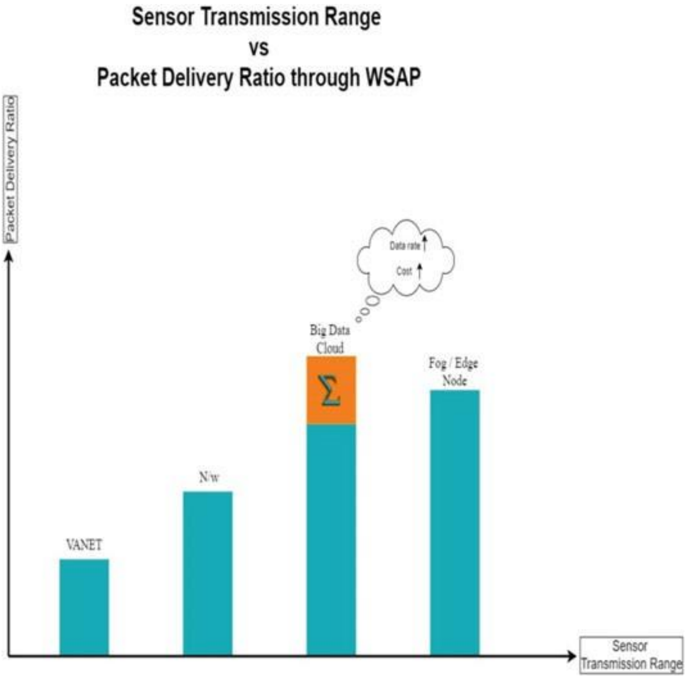


Fig. 9 Data transmission rate versus Fog-node packet delivery at wireless access points

8 Open Challenges on Future Fog-Era

Comprehensive work leaves the bellow proposals for the future research works that can be carried in emerging areas relying on the sustainable growth for smart cities,

1. To integrate the Fog Node-Fog Node interface communication collaborating the Inter-cloudlets for future application areas.
2. To expand the Fog-Nodes accessibility from remote server rather than transmitting the alert through the local node.
3. To realize the efficiency by interfacing fog-node for emergency recovery zones like Military Applications, Fire-monitoring in Forest, Industries, Patient-monitoring systems at minimal cost, minimal bandwidth, increased efficiency.
4. In predict the down-time of the Fog-Node and overcome the load-balancing during higher work load and available bandwidth.
5. To include more customizable working modes on Sensor execution apart from Relax-mode and Quick-mode as proposed in the active sensor modules.
6. To prove statistically the data rate speed, energy efficiency through simulator/tool kits.
7. To realize the fog-paradigm with a novel algorithm for Drone-applications, Robots, industrial applications, agriculture, weather forecasting, etc.

9 Conclusion

A Novel-taxonomy-PerColNif-algorithm is proposed, that clones the dataset from the source server $Point_A$ through two customizable Modes. This dynamic scheme ensures optimal energy spent on perceptual manifestation executed to process the central cloud, since the communication is through the glue of Fog-Node that clones only the collided incident from TLS on ADT thus results in low latency time, agile cloning achieved.

The work is simulated using Arduino board and real-time sensors equipped with vibrating and GPS sensors to realize location awareness for proving the fog-node as local-node bridging the Accident Detection Terminal with Hospital Server, these inter-connected services is not been discussed in any other existing works that might bring the evolving services in future technology.

References

1. Xu Y, Mahendran V, Radhakrishnan S (2016) Towards SDN-based fog computing: MQTT broker virtualization for effective and reliable delivery. In: 8th International conference on communication systems and network, pp 1--6. Bangalore, India
2. Bangui H, Rakrak S, Raghay S, Buhnova B (2018) Moving to the edge-cloud-of-things: recent advances and future research directions. Electronics 7:309
3. Isa ISM, Musa MOI, El-Gorashi TEH, Lawey AQ, Elmirghani JMH (2018) Energy efficiency of fog computing health monitoring applications. In: 20th International conference on transparent optical networks, pp 1--5. Bucharest
4. Osanaiye O, Chen S, Yan Z, Lu R, Choo KR, Dlodlo M (2017) From cloud to fog computing: a review and a conceptual live VM migration framework. IEEE Access 5:8284–8300
5. Tang B, Chen Z, Hefferman G, Wei T, He H, Yang Q (2015) A hierarchical distributed fog computing architecture for big data analysis in smart cities. In: Proceedings of the ASE big data & social informatics, pp 1--6. New York
6. Mukherjee M et al (2017) Security and privacy in fog computing: challenges. IEEE Access 5:19293–19304
7. Liu J et al (2018) Secure intelligent traffic light control using fog computing. Future Gener Comput Syst 78:817–824
8. Mohammadi M et al (2018) Deep learning for IoT big data and streaming analytics: a survey. IEEE Commun Surv Tutorials 20:2923–2960
9. Basudan S, Lin X, Sankaranarayanan K (2017) A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing. IEEE Internet Things J 4:772--782
10. Ajay Krishnan R, Magesh Kumar P, Vignesh R, Kesavan R, Veera Lakshmi P (2018) An automated design for ATC (Air Traffic Control) for airport monitoring system using TORADEX single board computer. Int J Res Appl Sci Eng Technol 6:2259–2267
11. Jiang Y, Chen Y, Yang S, Wu C (2019) Energy-efficient task offloading for time-sensitive applications in fog computing. IEEE Syst J 13:2930–2941
12. Kamal MB, Javaid N, Naqvi SAA, Butt H, Saif T, Kamal MD (2018) Heuristic min-conflicts optimizing technique for load balancing on fog computing. In: Xhafa F, Barolli L, Greguš M (eds) Advances in intelligent networking and collaborative systems. Lecture notes on data engineering and communications technologies, vol 23. Springer, Cham

13. Huo Y, Hu C, Qi X, Jing T (2017) LoDPD: a location difference-based proximity detection protocol for fog computing. IEEE Internet Things J 4:1117–1124
14. Joshi AY, Khanvilkar PS (2020) An energy efficient workload offloading in fog computing. Int Res J Eng Technol 7:1–6
15. Isa ISBM, El-Gorashi TEH, Musa MOI, Elmirghani JMH (2020) Energy efficient fog-based healthcare monitoring infrastructure. IEEE Access 8:197828–197852
16. Jo C, Gustafsson E, Son J, Egger B (2013) Efficient live migration of virtual machines using shared storage. In: 9th international conference on virtual execution environment, pp 4150, United State
17. Stojmenovic I, Wen S, Huang X, Luan H (2016) An overview of Fog computing and its security issues. Concurrency Comput: Pract Exp 28:2991–3005
18. Harish G, Nagaraju S, Harish B, Shaik M (2019) A review on fog computing and its applications. Int J Innov Technol Explor Eng 8:274–278
19. Dastjerdi AV, Gupta H, Calheiros RN, Ghosh SK, Buyya R (2016) Fog computing: principles, architectures, and applications. Int Things Principles Paradigms 61–75
20. Shi Y, Ding G, Wang H, Roman HE, Lu S (2015) The fog computing service for health care. In: 2nd IEEE international symposium on future information and communication technologies for ubiquitous healthcare, pp 1–5, China

Performance Evaluation and Energy Efficient VM Placement for Fog-Assisted IoT Environment

Sudhansu Shekhar Patra, Mamta Mittal, D. Jude Hemantha, Mahmoud A. L. Ahmad, and Rabindra Kumar Barik

Abstract The uses of Internet of Things (IoT) devices and sensors have been increasing day by day. In order to provide storage and computational needs for time-sensitive applications low powered end devices which use IoT devices and sensors, a new computing paradigm "Fog Computing" has come into the picture. Virtual machines (VMs) inside the fog nodes are responsible for immediate processing and analyzing the IoT workloads. One of the open research problems is to efficiently scalable the fog centers so that a minimum number of client requests can be renege from the fog system. The service providers are intended to retain the client requests in the fog system by providing efficient services. In this chapter, a multi-server queuing system having reneging with retention policy is modeled to measure the several performance measures of the fog system. The profit and revenue of the system are analyzed. Further, an efficient greedy-based VM placement scheme GVMP is proposed to optimize the energy consumption of the fog centers. The efficiency of the algorithm GVMP is compared with the state of art algorithms such as FFD, BFD, RR and MBFD.

Keywords Fog computing · IoT · Reneging · Retention · Queuing model · VM placement

S. S. Patra (✉) · R. K. Barik
School of Computer Applications, KIIT Deemed to be University, Bhubaneswar, India

M. Mittal
Department of Computer Science and Engineering, G. B. Pant Government Engineering College, New Delhi, India

D. Jude Hemantha
Department of Electronics and Communication Engineering, Karunya University, Coimbatore, Tamil Nadu, India

M. A. L. Ahmad
School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar, India

R. Tiwari et al. (eds.), *Energy Conservation Solutions for Fog-Edge Computing Paradigms*, Lecture Notes on Data Engineering and Communications Technologies 74,
https://doi.org/10.1007/978-981-16-3448-2_7

1 Introduction

The emergence of cloud computing is one of the technological paradigms and increasing day by day and is being used for processing IoT traffic because of the on-demand service, operational efficiency, scalability and energy efficiency. In the year 2021, around 50 billion IoT devices are connected to the Internet [1], and it will increase day by day. For IoT platforms, cloud computing is not the preferred platform due to its latency issues. To overcome this, fog or edge computing has been evolved in which the fog devices are kept nearby proximity of the IoT devices. Fog computing is a virtualized platform that renders storage, computation as well as networking services placed nearer to the IoT devices [1, 2]. Figure 1 shows the computing architecture of IoT applications and the fog computing layer.

Fog nodes are responsible for immediate processing and analyzing the IoT workloads and outcomes in notable performance. One of the major issues of a fog center is the scalability. It reduces the number VMs and in turn number of fog nodes which are allocated based upon IoT workload along with satisfying the QoS and SLA parameters. Upon arriving at the fog system, the client requests to join a queue, and if any one of the VM is free, the request is immediately processed; otherwise, it has to stay in the waiting queue. Most of the time either when the waiting time is more some client request leaves the queue and are abandoned from getting service or by looking at the number of requests waiting in the queue most of the client requests reneged. To gain the revenue, the provider has to retain them into the system. To model this, a finite multi-server queuing model with homogeneous VMs has been implemented which can help the service providers to analyze the number of VMs and the waiting buffer to be implemented in the system.

Energy efficiency is another QoS parameter of the fog center [4, 5]. Due to the increase in the use of the fog nodes, there is an increase in carbon footprints in the fog

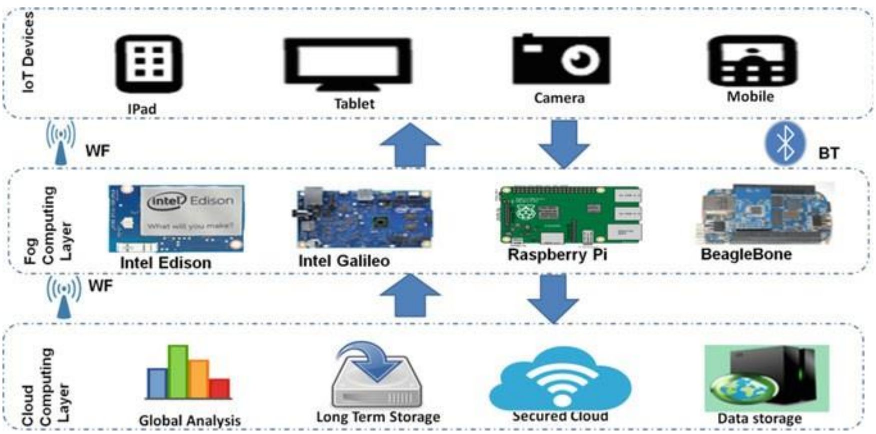


Fig. 1 Computing architecture of IoT applications and fog computing layer [3]

centers [6]. Therefore, the objective of the fog centers is to use the resources efficiently without compromising the client requirements to achieve green computing. The user requirements are given in the figure of the VM that is going to be deployed into the physical machines (PMs). This procedure helps in making the unused PMs to be switched off [7, 8] and so the energy consumption is reduced drastically.

1.1 Motivation

Many articles have been published on fog computing, but they have not focused on how the profitability of the service provider has been affected by the retention of the client requests because of the long waiting time in the system. The major contribution and the novelty of this chapter are as follows:

- To do the profit and revenue analysis of the system, a finite multi-server queuing system is implemented in which the system parameters are studied depending on the probability of retention of the client request.
- To optimize the energy consumption in the fog center, a greedy-based energy optimization algorithm has been implemented for VM placement in the fog nodes.

The rest of the chapter is organized as follows. Section 2 shows the previous research works, and Sect. 3 depicts the system model. Section 4 depicts the performance measures of the. The profit and revenue analysis of the system is done in Sect. 5. Section 6 gives the energy model of the fog system. Finally, Sect. 7 concludes the chapter.

2 Related Work

Cloud computing is a computing paradigm started from the last decade where the cloud provider has the ability to provide software, infrastructure and platform as a service on a pay per usage basis. Today, there are many such cloud providers which are engaged in providing their service to the customers including Google, Amazon web services (AWS), Microsoft and IBM offering innovative services to their clients. Due to latency issues in the cloud computing, the edge as well as fog computing came into picture. The performance analysis of the cloud system has been studied in [9] with queue dependent VMs. The profit analysis of M/M/c queuing systems with balking and reneging is studied in [10]. In the fog centers, the idle resources consume a significant amount of power. Optimization of energy consumption is a critical issue in the cloud data centers. There is much research going on in this field for optimizing energy. Many practitioners [11–14] have proposed many VM placement algorithms for minimizing the energy consumption in the data centers. In [15–17], the authors surveyed on server consolidation, and they observed that the major issues of consolidation are inefficient VM placement, frequent migration of VMs and SLA violation. In [18], the authors have suggested MBFD which improves the quality of service. The authors in [19, 20] have suggested a dynamic round robin (RR) algorithm for the server consolidation. In [21], authors have used genetic algorithms and best fit strategy to minimize the energy used by the minimization of the PMs in the current state and by the reduction of resource wastage. In many literature [22–24], the authors used the weighted PageRank to reduce the number of PMs. In [25], authors have used MILP along with branch and bound methods including three heuristics that minimizes the energy consumption. The heuristic algorithms are efficient but the solution the algorithms generated is not promising. In [26], principles of energy conservation for IoT have been discussed.

3 System Model

The proposed model of fog assisted IoT environment is shown in Fig. 2. The client requests have to wait in the queue before being processed by the fog nodes. If the VMs inside the fog nodes are free, then the client requests are served immediately; otherwise, they have to wait in the queue. In some situations, if the client requests have to wait for a long time after joining the queue, the requests may leave the queue, or by looking at the queue size, the client request may renege. But the service providers always try to retain the impatient client requests. A reneged client request may be retained in the waiting buffer for the service with a probability of q and not retained with probability p (=1−q), i.e., the client abandons the queue. The assumptions made are as follows:

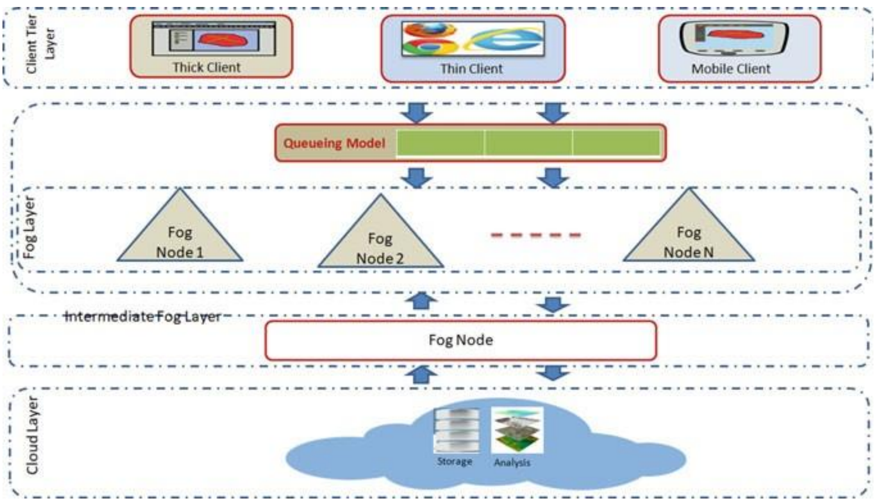


Fig. 2 Fog-assisted IoT environment with queuing model

1. The arrival rate of the client requests is λ, and the requests are coming to the fog system which follows the Poisson distribution.
2. There are r VMs in the system, and the service times of each VM follow an exponential distribution with parameter μ. The mean service rate is denoted by:
$$\mu_k = \begin{cases} k\mu; \ 0 \le k \le r-1 \\ r\mu; \ r \le k \le N \end{cases}$$
3. The client requests are served by the VMs using FCFS discipline.
4. The capacity of the waiting buffer is S, i.e., the system can accommodate at most S requests.
5. The client requests join the waiting buffer as and when the client requests exceed the number of VMs, i.e., when $k > r$. Each client requests upon joining a waiting buffer will wait for some time to get the service. If within a certain length of time the service has not been started, the customer will get impatient or renege the service and may leave the waiting queue without getting service having probability p or may wait in the waiting buffer to get service with probability q. For reneging, the distribution is considered to be exponential with parameter γ.

The difference equations for the system can be written as:

$$\frac{\mathrm{d}\pi_0(t)}{\mathrm{dt}} = -\lambda\pi_0(t) + \mu\pi_1(t)\,;\ k = 0 \tag{1}$$

$$\frac{\mathrm{d}\pi_k(t)}{\mathrm{dt}} = -(\lambda + k\mu)\pi_k(t) + (k+1)\mu\pi_{k+1}(t) + \lambda\pi_{k-1}(t)\,;\ 1 \le k \le r-1 \tag{2}$$

$$\begin{aligned}\frac{\mathrm{d}\pi_k(t)}{\mathrm{dt}} &= -[\lambda + c\mu + (k-c)\gamma p]\pi_k(t) \\ &\quad + [c\mu + \{(k+1) - c\}\lambda p]\pi_{k+1}(t) + \lambda\pi_{k-1}(t); \\ &\quad r \le k \le S-1\end{aligned} \tag{3}$$

$$\frac{\mathrm{d}\pi_S(t)}{\mathrm{dt}} = \lambda\pi_{S-1}(t) - \{r\mu + (S-r)\gamma p\}\pi_S(t);\ k = S \tag{4}$$

No client request is in the waiting queue and so no chance of reneging as shown in Eqs. (1) and (2). Equations (3) and (4) deal with the situations, whenever the number of client requests is greater than the number of VMs.

In steady state, $\lim_{k \to \infty} \pi_k(t) = \pi_k$. So, the equations in the steady state from (1) to (4) are as follows:

$$0 = -\lambda\pi_0 + \mu\pi_1 \tag{5}$$

$$0 = -(\lambda + k\mu)\pi_k + (k+1)\mu\pi_{k+1} + \lambda\pi_{k-1}\,;\ 1 \le k \le r-1 \tag{6}$$

$$0 = -[\lambda + c\mu + (k - r)\gamma\, p]\pi_k + [c\mu + \{(k + 1) - c\}\lambda p]\pi_{k+1} + \lambda\pi_{k-1};\ r \le k \le S - 1 \tag{7}$$

$$0 = \lambda\pi_{S-1} - \{c\mu + (S - r)\gamma\, p\}\pi_S;\ k = S \tag{8}$$

The recursive solution from Eqs. (5) to (8) gives,

$$\pi_k = \frac{\lambda^k}{k!\mu^k}\pi_0;\ 1 \le k \le r \tag{9}$$

$$\pi_k = \prod_{l=r+1}^{k} \frac{\lambda}{r\mu + (l - r)\gamma\, p} \frac{\lambda^r}{r!\mu^r}\pi_0;\ r + 1 \le k \le S \tag{10}$$

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{r} \frac{\lambda^k}{k!\mu^k} + \sum_{k=r+1}^{S} \prod_{l=r+1}^{k} \frac{\lambda}{r\mu + (l-r)\gamma\, p} \frac{\lambda^r}{r!\mu^r}} \tag{11}$$

4 Performance Measures

System performance has a great impact on the stability of any system. The expected length of the fog system, the expected buffer size, average waiting time of a client request in the fog system, expected waiting time of a client request in the waiting buffer are the various performance measures of the fog system and can be calculated as follows:

1. The expected length of the fog system (L_s):

$$L_s = \sum_{k=0}^{S} k P_k \tag{12}$$

$$L_s = \left[\sum_{k=1}^{r} k \frac{\lambda^k}{k!\mu^k} + \sum_{k=r+1}^{S} k \prod_{l=r+1}^{k} \frac{\lambda}{r\mu + (l - r)\gamma\, p} \frac{\lambda^r}{r!\mu^r}\right]\pi_0 \tag{13}$$

2. The expected buffer length (L_q):

$$L_q = L_s - \frac{\lambda}{\mu} = \left[\sum_{k=1}^{r} k \frac{\lambda^k}{k!\mu^k} + \sum_{k=r+1}^{S} k \prod_{l=r+1}^{k} \frac{\lambda}{r\mu + (l - r)\gamma\, p} \frac{\lambda^r}{r!\mu^r}\right]\pi_0 - \frac{\lambda}{\mu} \tag{14}$$

3. The average waiting time in the fog system (W_s)

$$W_s = \frac{L_s}{\lambda} = \frac{\left(\sum_{k=1}^{r} k \frac{\lambda^k}{k!\mu^k} + \sum_{k=r+1}^{S} k \prod_{l=r+1}^{k} \frac{\lambda}{r\mu + (l-r)\gamma p} \frac{\lambda^r}{r!\mu^r}\right)\pi_0}{\lambda} \tag{15}$$

4. The average waiting time in the waiting buffer (W_q)

$$W_q = \frac{L_q}{\lambda} = \frac{\left(\sum_{k=1}^{r} k \frac{\lambda^k}{k!\mu^k} + \sum_{k=r+1}^{S} k \prod_{l=r+1}^{k} \frac{\lambda}{r\mu + (l-r)\gamma p} \frac{\lambda^r}{r!\mu^r}\right)\pi_0}{\lambda} - \frac{1}{\mu} \tag{16}$$

where

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{r} \frac{\lambda^k}{k!\mu^k} + \sum_{k=r+1}^{S} \prod_{l=r+1}^{k} \frac{\lambda}{r\mu + (l-r)\gamma p} \frac{\lambda^r}{r!\mu^r}} \tag{17}$$

Table 1 shows the different measures of effectiveness with various probabilities of retaining the impatient client retests. The parameters in the study are taken as $\mu = 3$, $\lambda = 2$, $S = 4$, $\gamma = 0.1$ and $r = 2$. One can observe from Table 1 that as the increased probability of retaining the client request to the fog system increases, L_s, L_q, W_s and W_q increases steadily. $q = 0$ means that there is no client request retained to the queue, and there will be only reneging. But when $q = 1$, means there are no client requests, they are impatient, and so all the performance measures are maximum.

5 Profit and Revenue Analysis of the System

The utility of any system enhances with the economic analysis such as cost-profit analysis of the system. This section gives an overall analysis to the providers the expected cost, revenue and total profit of the system. The various parameters involved in the system are mentioned in Table 2.

In a finite capacity fog system, some client requests may leave the system when the system is full and let the rate at which they lost the system is λ_{lost} and can be expressed as:

$$\lambda_{\text{lost}} = \lambda \times \pi_s \tag{18}$$

where

Table 1 Different performance measures W_s, W_q, L_s and W_q by varying the probability of retention of an impatient client request (q)

$q(=1-p)$	W_s	W_q	L_s	L_q
0	0.35736	0.025237	0.71723	0.050347
0.05	0.35823	0.025342	0.71745	0.050456
0.1	0.35927	0.025367	0.71982	0.050672
0.15	0.36352	0.025419	0.72019	0.050879
0.2	0.36382	0.025576	0.72346	0.051045
0.25	0.36879	0.025597	0.73567	0.051372
0.3	0.37282	0.025624	0.73712	0.051568
0.35	0.37829	0.025737	0.73817	0.051873
0.4	0.38234	0.025827	0.74102	0.052032
0.45	0.38527	0.025929	0.74342	0.052365
0.5	0.38928	0.026123	0.74452	0.052563
0.55	0.39124	0.026314	0.74546	0.052762
0.6	0.39483	0.026562	0.74637	0.052892
0.65	0.39798	0.026678	0.74873	0.052983
0.7	0.39982	0.026979	0.75027	0.053127
0.75	0.40012	0.027123	0.75342	0.053256
0.8	0.40437	0.027345	0.75567	0.053412
0.85	0.40462	0.027562	0.75768	0.053528
0.9	0.40782	0.027675	0.75982	0.053629
0.95	0.40982	0.027789	0.76105	0.053872
1	0.41352	0.027982	0.76349	0.053984

$$\pi_S = \prod_{l=r+1}^{S} \frac{\lambda}{r\mu + (l-r)\gamma p} \frac{\lambda^r}{r!\mu^c} \pi_0$$

The mean reneging rate R_{Ren} and the mean retention rate R_{Ret} can be calculated as:

$$R_{\mathrm{Ren}} = \sum_{k=1}^{S} (k-r)\gamma p \pi_k \tag{19}$$

$$R_{\mathrm{Ret}} = \sum_{k=1}^{S} (k-r)\gamma q \pi_k \tag{20}$$

Expected cost can be defined as:

Table 2 Different parameters used in the system

Parameters	Meaning
$1/\mu$	Client request mean service time
$1/\lambda$	Client request inter-arrival time
$\rho = \lambda/\mu$	Traffic intensity of the system to know how much the system is busy
λ_{lost}	Rate at which the client requests are lost
P_S	Probability the system being full
L_s	Expected no. of client requests in the fog system
R_{Ret}	Rate of retention
R_{Ren}	Rate of reneging
C_{service}	Service cost per unit time
C_{holding}	Holding cost of a client request per unit time
C_{lost}	Cost per unit time of losing a client request
C_{Ren}	Cost per unit time associated with each reneged client request
C_{Ret}	Cost per unit time associated with each retained client request
Rev	Revenue earned by providing service to the client request

$$\text{Expected Cost} = C_{\text{service}}\mu + C_{\text{holding}} L_s + C_{\text{lost}}\lambda\,\pi_s + C_{\text{Ren}} R_{\text{Ren}} + C_{\text{Ret}} R_{\text{Ret}} \quad (21)$$

where

$$L_s = \sum_{k=1}^{r} k \frac{\lambda^k}{k!\mu^k} + \sum_{k=r+1}^{S} k \sum_{l=r+1}^{k} \frac{\lambda}{r\mu + (l-r)\gamma p} \frac{\lambda^r}{r!\mu^r} \pi_0 \quad (22)$$

The expected revenue of the system can be calculated by:

The total revenue earned by rendering service to average client requests minus the sum of loss in revenue of the fog system due to the loss of client request/time and due to the revenue loss due to the reneging of client request/time in the fog system. Expected revenue can be expressed as follows:

$$\text{Expected Revenue} = (\text{Rev} * L_s) - (\text{Rev} \times \lambda \times \pi_s) - (\text{Rev} \times R_{\text{Ren}}) \quad (23)$$

The total expected profit can be expressed as:

$$\text{Expected Profit} = \text{Expected Revenue} - \text{Expected cost} \quad (24)$$

So,

Table 3 TER, TEC and TEP by varying the probability of retention of an impatient client request (q)

$q(=1-p)$	TER	TEC	TEP
0	202.41323	69.94002	132.47291
0.05	202.77262	69.94538	132.82723
0.1	203.34521	70.02522	133.24526
0.15	203.73638	70.36273	133.81628
0.2	204.12763	70.78374	134.25276
0.25	204.36383	71.12783	134.62822
0.3	204.82729	71.46376	134.93783
0.35	205.12455	71.96879	135.32455
0.4	205.47273	72.37483	135.93739
0.45	205.92623	72.82723	136.26276
0.5	206.23244	73.13484	136.67637
0.55	206.87335	73.36347	136.92733
0.6	207.24574	73.72648	137.37272
0.65	207.64758	73.93748	137.83738
0.7	207.84276	74.34162	138.25262
0.75	208.13233	74.74839	138.82929
0.8	208.63829	75.13823	139.25262
0.85	208.89276	75.47628	139.78292
0.9	209.23151	75.82028	140.14244
0.95	209.82728	76.25262	140.62938
1	210.23133	76.83903	140.93739

$$\begin{aligned}\text{Expected Profit} &= (\text{Rev} * L_s) - (\text{Rev} \times \lambda \times \pi_s) - (\text{Rev} \times R_{\text{Ren}}) \\ &\quad - (\text{C}_{\text{service}}\mu + \text{C}_{\text{holding}} L_s + \text{C}_{\text{lost}}\lambda\, \pi_s + C_{\text{Ren}} R_{\text{Ren}} + C_{\text{Ret}} R_{\text{Ret}}) \\ &= (\text{Rev} - C_{\text{holding}}) * L_s - (\text{Rev} + \text{C}_{\text{lost}}) * \lambda \times \pi_s \\ &\quad - (\text{Rev} - C_{\text{Ren}}) R_{\text{Ren}} - C_{\text{service}}\mu - C_{\text{Ret}} R_{\text{Ret}} \end{aligned} \tag{25}$$

Table 3 shows the total expected revenue (TER), total expected cost (TEC) and total expected profit (TEP) based on the probability of retention of impatient clients.

From Table 3, one can see that with the increase in retaining the client requests, there will be an increase in total revenue of the fog system. The total expected profit also increases as q increases. This helps the service providers the retention policies by knowing the profits earned with the involved costs. The values taken in computing Table 3 are $\mu = 3, \lambda = 2, S = 4, \gamma = 0.1, r = 2, C_{\text{service}} = 20, C_{\text{hold}} = 10, C_{\text{lost}} = 25, C_{\text{Ren}} = 8, C_{\text{Ret}} = 10$ and $\text{Rev} = 300$.

6 The Energy Model of the Fog System

A fog node has a set of VMs, m PMs with m hypervisors and management entities and different sets of switches shown in Fig. 3. Each PM has the capabilities of hosting multiple VMs. A hypervisor in a PM is capable of managing a set of VMs. In Pods, the PMs are grouped. A pod consists of multiple switches located inside a rack and is connected to each server. The role of the allocation manager is to place the VM's by fairly distributing the VMs over the PMs so that the VMs can receive the necessity resources such as CPU and memory requirements from the PM hypervisor.

6.1 VM Placement Problem Formulation

The VM placement problem can be mathematically formulated as follows:

1. Physical machines: Let PM={PM_1, …, PM_m} be the set of m heterogeneous PMs.
2. Virtual machines: Let VM={VM_1, VM_2, …, VM_n} be a set of n heterogeneous VMs.
3. VMP problem: When a client requests a service, the fog layer finds a suitable VM based on the capacity of the request. However, several requests can be placed to the same VM. The VMs are heterogeneous. The used VMs are placed in the PMs of the fog nodes. Multiple VMs can be placed in the same PMs of the fog node. This problem can be represented by a mapping function $f: VM \rightarrow PM$ in such a way that power optimization can be achieved. Let $M_{m \times n}$ be a binary mapping matrix to represent the mapping function f between VMs and PMs in which each element is defined as:

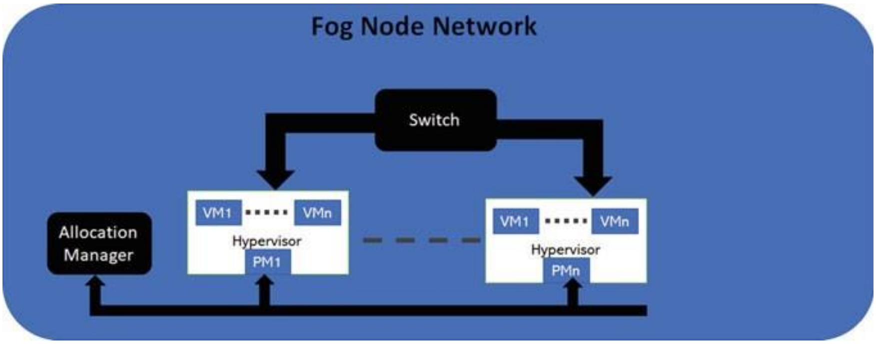


Fig. 3 Fog node network

$$m_{ij} = \begin{cases} 1 \text{ if VM}_j \text{ is placed on PM}_i \\ 0 \text{ otherwise } \forall i \in \text{VM}, \ \forall j \in \text{PM} \end{cases} \tag{26}$$

The allocation vector A_n identifies the PM's state, where its one element entry will be one when at least one VM is hosted on the corresponding PM, otherwise it will be zero. The allocation vector is represented as follows:

$$a_i = \begin{cases} 1 \quad \text{if } \sum_{j=1}^{n} m_{ij} \geq 1 \\ 0 \text{ otherwise } \forall i \in \text{PM} \end{cases} \tag{27}$$

We assume that the requested information, VM's capacity and PM's capacity all are in the form of million instructions per second (MIPS),, and they noted before the placement taking place.

The resources which are measurable and consume power in a datacenter are processor or CPU, network and storage. However, the power consumption by a processor is greater than the other components. The processor usage depends on the load of the system [5]. An idle server consumes roughly around 70% of the power consumption in comparison with a fully utilized server [5, 7]. Thus, there is a need to switch off the idle servers or put them in sleep mode to reduce significant power. The power model of a server (Pow) is mathematically expressed as follows:

$$\text{Pow}(U) = \begin{cases} \text{Pow}_{\min} + (\text{Pow}_{\max} - \text{Pow}_{\min}) \times U \text{ if } U > 0 \\ 0 \text{ otherwise} \end{cases} \tag{28}$$

where $\text{Pow}_{\min}$ is the power consumption of an idle server (i.e., 70%), $\text{Pow}_{\max}$ is the power consumption by a fully utilized server, and U is the utilization of the processor.

Often, the utilization of the CPU varies with respect to time because of the workload variability. So the CPU utilization represented by $U(t)$ is a function of time. Thus, the total energy consumption of a physical node (E) is defined as a definite integral of the power consumption function over a period of time.

The energy consumed for a period $[a, b]$ is as follows:

$$E = \int_a^b \text{Pow}(U(t))\text{d}t \tag{29}$$

6.2 *Greedy Heuristic Algorithm for Energy Saving*

The objective of the algorithm is two folds.

1. Minimize the power consumption of the fog network.
2. Minimization of VMs as well as PMs.

Algorithm: **GREEDY VM _Placement (GVMP)**

Algo Greedy_VM_Placement (CU,CV,CP)
Input: A set U of user requests and CU[1..m] is the load of m user requests in terms of MIPS
A set V of VMs and CV[1..n] is the capacity of n VMs in terms of MIPS
A set P of PMs and CP[1..k] is the capacity of k PMs in terms of MIPS
Output:
1. Allocation of client requests to VMs
2. Allocation of VMs to PMs

Stage-1: Allocation of client requests to VMs

Step 1: Sort the set U in decreasing order based on CU array
Step 2: Sort the set V in decreasing order based on CV array
Step 3: Call **Greedy_Allocation** (U set of client requests, set of VMs, m number of User requests, n number of VMs, CU capacity array of client requests, CV Capacity array of VMs) //the output from this procedure is the set V' where $V' \subseteq V$ along with CV'

Stage-2: Allocation of VMs to PMs

Step 1: Sort the set P in non-increasing order based on CP array
Step 2: Call **Greedy_Allocation** (V' is the set of allocated VMs, P set of PMs, n' number of allocated VMs, k number of PMs, CV' capacity array of allocated VMs, CP capacity array of PMs) //the output from this procedure is the set P' where $P' \subseteq P$ along with CP'

Procedure **Greedy_Allocation** (A, B, p, q, CA, CB)

//This algorithm did the allocation of p number of objects stored in array A with the capacity array CA in q number of objects stored in array B with capacity array CB.
1. for j=1 to d do
2. set allocate_xm[j]=Empty //Initialize all the machines are Empty
3. for i=1 to p do
4. for j= 1 to q do
5. if ((allocate_xm[j]= =0) && (CB[j] >= CA[i]))
6. Allocate X_i to Y_j
7. CBj]=CB[j]-CA[i]
8. if (CB[j]= =0)
9. Allocate_xm[j]=1
10. break
11. return

6.3 An Illustration

Let us take an illustration to understand the algorithm consisting of 6 client requests, 6 VMs and 3 PMs. The client requests are <U1, U2, U3, U4, U5, U6> with the capacity array CU <150, 200, 300, 350, 150, 300>, the VMs are <V1, V2, V3, V4, V5, V6> with the capacity array CV <1000, 600, 500, 700, 250, 400>, and the PMs are <P1, P2, P3> with the capacity array CP <1700, 850, 900>. By applying the above Greedy_VM_Placement, the sorted order of U, V and P are <U4, U3, U6, U2, U1, U5>, <V1, V4, V2, V3, V6, V5> and <P1, P3, P2>, respectively. The allocation for the client requests to the VMs is shown in Table 4, and the allocation of VMs to PMs is shown in Table 5. Note that after the allocation of client requests to the VMs, the sorted order of VMs V^r in terms of CV^r are <V1, V2, V3, V4, V6, V5> and <1450, 600, 500, 500, 400, 250>, respectively. The utilization of P1 is 1450 out

Table 4 Allocation of client requests to VMs

Client request	Allocated to VM	MIPS of client request
U4	V1	350
U3	V1	300
U6	V1	300
Total capacity allocated in VM	V1	950
U2	V4	200
U1	V4	150
U5	V4	150
Total capacity allocated in VM	V4	500

Table 5 Allocation of VMs to PMs

VM request	Allocate to PM	MIPS of VM request
V1	P1	950
V4	P1	500
Total capacity allocated in PM	P1	1450
V2	P2	600
V5	P2	250
Total capacity allocated in PM	P2	850
V3	P3	500
V6	P3	40
Total capacity allocated in PM	P3	900

of the maximum capacity of 1700 with 85.29%, and the energy consumption would be 70% of 250 + 30% of 250 * 85.29 = 652 units by considering Pow_{max} = 250 units and Pow_{min} = 70% of P_{max} because the power utilization by idle server is 70% of Pow_{max} [18].

6.4 Simulation Results

6.4.1 Dataset Description and Simulation Setup

The simulation has been done using Python 3.7 on an Intel ® Core ™ i5-8250U CPU @ 1.60 GHz and 8 GB RAM on Windows 10 Home 64-bit OS. The performance of the algorithm has been studied with some generated datasets. The structure of the datasets used is ix nCR × nVM × nPM, where i stands; for the instance number, nCR, nVM and nPM represents the number of client requests, number of VMs and number of PMs, respectively. There are five instances, i.e., i1–i5 have been run during the simulation varying the number client requests nCR from 100 to 10,000, number of VMs nVM from 10 to 1000 and number of PMs nPM from 5 to 500.

6.4.2 Results Analysis

The algorithm GVMP is compared in different performance metrics such as VMs used (M1), PMs used (M2), avg. PM utilization (M3) and energy consumption (M4) with various state of art algorithms such as FFD, RR, BFD, MBFD which is shown in Tables 6 and 7. M3 is computed as sum (utilization of PMs)/no. of PMs. The pictorial representation of energy consumption is shown in Fig. 4 which shows the energy consumption is less in GVMP in companion to all other algorithms.

Table 6 Comparison of performance matrices M1 and M2 for various algorithms

Dataset	FFD		BFD		RR		MBFD		GVMP	
	M1	M2	M1	M2	M1	M2	M1	M2	M1	M2
100 × 10 × 5	9	4	7	6	9	4	9	4	8	3
500 × 50 × 25	48	16	45	23	47	18	48	24	35	12
1000 × 100 × 50	100	29	98	38	88	48	100	28	88	26
5000 × 500 × 250	500	143	433	238	486	190	490	248	320	68
10,000 × 1000 × 500	1000	288	1000	368	875	480	1000	287	670	145

Table 7 Comparison of performance matrices M3 and M4 for various algorithms

	Datasets					
Algorithm	Parameter	100 × 10 × 5	500 × 50 × 25	1000 × 100 × 50	5000 × 500 × 250	10,000 × 1000 × 50
FFD	M3	80	90	91	94	93
	M4	2.26e+04	1.1e+05	1.97e+05	1.04e+06	2.05e+06
BFD	M3	83	92	92	95	95
	M4	2.65 e+04	1.33e+05	2.61e+05	1.35e+06	2.66e+06
RR	M3	58	57	54	56	56
	M4	2.13e+04	1.07e+05	1.99e+05	1.04e+06	2.07e+06
MBFD	M3	80	90	90	92	92
	M4	2.16e+04	1.08e+05	2.00e+05	1.02e+06	2.03e+06
GVMP	M3	68	88	92	95	94
	M4	1.19e+04	5.09e+04	9.10e+04	4.56e+05	9.10e+05

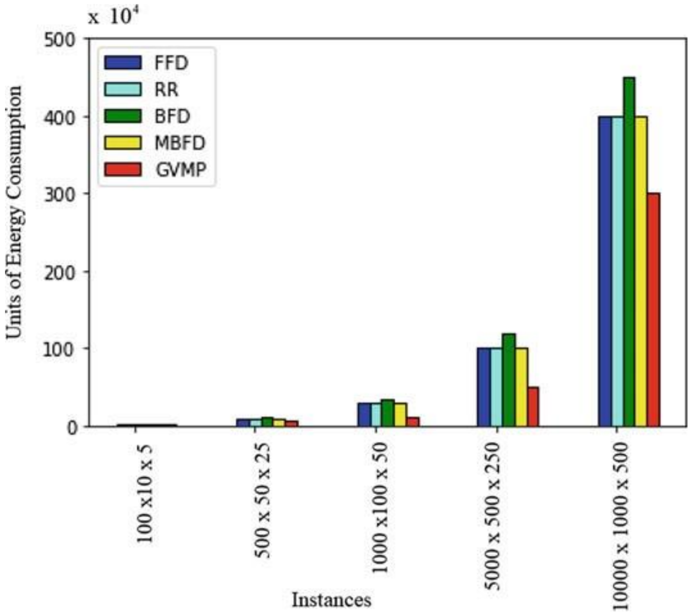


Fig. 4 Energy comparison of GVMP with the state of art algorithms

7 Conclusion

Fog nodes are in near proximity of the IoT devices as compared to the cloud servers. The VMs inside the PMs take the responsibility for immediate processing and analyzing the client requests. When all the VMs are busy serving the other client requests, the client waits in the waiting buffer until their turn to come. In some cases, the client requests leave the waiting buffer before being served. The service provider wants to retain the client requests by providing efficient service. In this chapter, various performance measures are measured by varying the probability of retention of impatient client requests. Also the profit and revenue analysis of the system is done. The total expected revenue, total expected cost and total expected profit are calculated by varying the probability of retention of impatient customers. A greedy-based heuristic algorithm.

GVMP has been implemented for VM placement in the fog nodes. The performance of the algorithm is compared with different state of art algorithms which perform well in energy saving in the fog nodes. In the future, machine learning algorithms have to be used for VM placement and consolidation in fog servers.

References

1. Al Ridhawi I, Kotb Y, Aloqaily M, Jararweh Y, Baker T (2019) A profitable and energy-efficient cooperative fog solution for IoT services. IEEE Trans Industr Inf 16(5):3578–3586
2. Barik RK, Dubey H, Mankodiya K, Sasane SA, Misra C (2019) GeoFog4Health: a fog-based SDI framework for geospatial health big data analysis. J Ambient Intell Human Comput 10(2):551–567
3. Fantacci R, Picano B (2020) Performance analysis of a delay constrained data offloading scheme in an integrated cloud-fog-edge computing system. IEEE Trans Veh Technol 69(10):12004–12014
4. Priyadarshini R, Barik RK, Dubey H (2018) Deepfog: fog computing-based deep neural architecture for prediction of stress types, diabetes and hypertension attacks. Computation 6(4):62
5. Al Ahmad M, Patra SS, Barik RK (2020) Energy-efficient resource scheduling in fog computing using SDN framework. In: Progress in computing, analytics and networking. Springer, pp 567–578
6. Al-Khafajiy M, Baker T, Al-Libawy H, Maamar Z, Aloqaily M, Jararweh Y (2019) Improving fog computing performance via fog-2-fog collaboration. Futur Gener Comput Syst 100:266–280
7. Isa ISBM, El-Gorashi TE, Musa MO, Elmirghani JM (2020) Energy efficient fog-based healthcare monitoring infrastructure. IEEE Access 8:197828–197852
8. Shaw R, Howley E, Barrett E (2021) Applying reinforcement learning towards automating energy efficient virtual machine consolidation in cloud data centers. Inf Syst 101722
9. Goswami V, Patra SS, Mund GB (2012) Performance analysis of cloud with queue-dependent virtual machines. In: 2012 1st international conference on recent advances in information technology (RAIT). IEEE, pp 357–362
10. Chauhan MS, Sharma GC (1996) Profit analysis of queueing model with reneging and balking. Monte Carlo Methods Appl 2:139–144
11. Yang Y, Wang K, Zhang G, Chen X, Luo X, Zhou MT (2018) MEETS: maximal energy efficient task scheduling in homogeneous fog networks. IEEE Internet Things J 5(5):4076–4087
12. Singh PP, Khosla PK, Mittal M (2019) Energy conservation in IoT-based smart home and its automation. Energ Conserv IoT Dev 155–177
13. Saraswat S, Gupta HP, Dutta T, Das SK (2019) Energy efficient data forwarding scheme in fog-based ubiquitous system with deadline constraints. IEEE Trans Netw Serv Manage 17(1):213–226

14. Shahid MH, Hameed AR, ul Islam S, Khattak HA, Din IU, Rodrigues JJ (2020) Energy and delay efficient fog computing using caching mechanism. Comput Commun 154:534–541
15. Saeedi P, Shirvani MH (2021) An improved thermodynamic simulated annealing-based approach for resource-skewness-aware and power-efficient virtual machine consolidation in cloud datacenters. Soft Comput 1–28
16. Qu Z, Wang Y, Sun L, Peng D, Li Z (2020) Study QOS optimization and energy saving techniques in cloud, fog, edge, and IOT. Complexity
17. Ahmad RW, Gani A, Hamid SHA, Shiraz M, Yousafzai A, Xia F (2015) A survey on virtual machine migration and server consolidation frameworks for cloud data centers. J Netw Comput Appl Elsevier. 52:11–25
18. Beloglazov A, Abawajy J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Futur Gener Comput Syst 28(5):755–768
19. Vilela PH, Rodrigues JJ, Solic P, Saleem K, Furtado V (2019) Performance evaluation of a Fog-assisted IoT solution for e-Health applications. Futur Gener Comput Syst 97:379–386
20. Lin C, Liu P, Wu J (2011) Energy-aware virtual machine dynamic provision and scheduling for cloud computing. In: 4th international conference on cloud computing. IEEE, pp 736–737
21. Patra SS (2018) Energy-efficient task consolidation for cloud data center. Int J Cloud Appl Comput (IJCAC) 8(1):117–142
22. Alharbi HA, Elgorashi TE, Elmirghani JM (2020) Energy efficient cloud-fog architecture. arXiv:2001.06328
23. Gougeon A, Camus B, Orgerie AC (2020) Optimizing green energy consumption of fog computing architectures. In: 2020 IEEE 32nd international symposium on computer architecture and high performance computing (SBAC-PAD). IEEE, pp 75–82
24. Singh R, Gehlot A, Mittal M, Samkaria R, Choudhury S (2017) Application of icloud and wireless sensor network in environmental parameter analysis. Int J Sens Wireless Commun Control 7(3):170–177
25. Wei C, Hu ZH, Wang YG (2020) Exact algorithms for energy-efficient virtual machine placement in data centers. Futur Gener Comput Syst 106:77–91
26. Mittal M, Pandey SC (2019) The rudiments of energy conservation and IoT. In: Energy conservation for IoT devices. Springer, pp 1–17

Load Balancing in Fog Computing Using QoS

Shilpi Harnal, Gaurav Sharma, Nidhi Seth, and Ravi Dutt Mishra

Abstract Various IoT gadgets are ceaselessly increasing day by day and producing an enormous volume of raw data. All of this produced information is passed to cloud servers for processing. As this process adds delay to processing, so it is not suitable for certain applications as some applications require a speedy response. To overcome this condition, fog computing comes in existent, which is an extension to cloud computing. Also, in the present time, fog is the most popular technology due to the vast demand for IoT devices. The fog nodes are placed between IoT devices and the cloud servers. As the execution of the request is performed at the fog layer so it can work with a limited number of resources i.e., less bandwidth, cost, and time as the processing is pushed closer to the end clients. The most challenging task in a fog environment is to appropriately distribute workload among computing nodes during the execution of IoT applications as it is one of the important factors which affect resource efficiency. The performance of any computing paradigm is directly proportional to the load balancing handling mechanism; poor mechanism reduces the overall performance of any computing environment. Realizing the challenge of load balancing among the computing nodes in the fog environment, various mechanisms and methods have been proposed so far and various experiments have also been conducted by the researchers to check the effectiveness of the mechanism. The appropriate load balancing mechanism will increase the effectiveness of the fog system due to better resource utilization. The chapter presents a framework (OLBA) for Load Balancing in Fog computing environments to balance the load between fog devices and improves QoS parameters i.e., Turnaround time resource utilization, response time, and delay parameter. This approach is based on Particle Swarm Optimization (PSO) technique to find the local best and then to compare all the local best to find the ultimate global best solution. An analysis and comparison with the traditional techniques, i.e., FCFS, SJF, Max_Min is also performed for a better understanding of load balancing mechanism in Fog Computing.

S. Harnal (✉)
Institute of Engineering and Technology, Chitkara University, Punjab, India
e-mail: shilpi.harnal@chitkara.edu.in

G. Sharma · N. Seth · R. D. Mishra
Seth Jai Parkash Mukand Lal Institute of Engineering and Technology, Radaur, Haryana, India

R. Tiwari et al. (eds.), *Energy Conservation Solutions for Fog-Edge Computing Paradigms*, Lecture Notes on Data Engineering and Communications Technologies 74,
https://doi.org/10.1007/978-981-16-3448-2_8

1 Introduction

In the past few years, we have seen vast development in the fields of distributed, grid, and cloud computing paradigms. Cloud computing renders on-demand resources as general utilities that can be leased and released by clients through the Internet. Cloud computing [1, 2] is on-demand availability of computer resources like data storage and data computing that has several inbuilt capabilities such as expandability, minimized management efforts, varying pricing models, allocation of resources, and easily available services and applications [3].

As in today's era, cloud computing is widely used in everyday life, it still suffer with some flaws [4]. The major flaw is the poor internet connection between the cloud and the end users, that means—if no internet then no cloud [5]. This limitation does not suit a large set of applications that are latency-sensitive [6], i.e., vehicles movement [7], detect fire [8], and content delivery [9]. Most of the cloud-based applications are distributed in nature and consist of several elements [10]. Therefore, it sometimes becomes difficult to deploy application elements distinctively over various clouds (e.g., [11, 12]). This inter-cloud correspondence may further harm the latency because of extra overhead. Another flaw is if the cloud host is not having data centers for storage locations mentioned by the procedures' computation [13].

Fogging [14] is a computing model recommended to handle such issues. These issues have now also been addressed by the Open Fog Consortium. The organization has also published some white papers (e.g., [15]) covering the critical issues. Fog can be seen as "a cloud closer to the ground". Fogging is an infrastructure which is distributed in nature and some applications mainly those which are latency-sensitive and many services also are managed nearer to the edge either by fog entities or using help of IoT devices [16] while those applications which are not latency-sensitive, are still managed in the cloud. Services such as processing, networking, and storing are the main components of the cloud layer and also of the fog layer that extends it. This fog layer permits the processing to take place near to the IoT devices which are assisted by fog entities and also enables processing at distinct locations. Usage of IoT devices like smartwatches and smartphones are increasing tremendously. All IoT devices produce immense amounts of data generated by the sensors to sense the real-time environment [17]. This data is processed by cloud servers. But some applications such as the military operations and health require a quick response where delay is critical [18]. Fog layer is placed between the cloud and IoT to overcome this delay. Fog servers process data faster and provide immediate response and essential data is passed to the layer of cloud computing. In fog computing computation, communication, and storage are held at edge devices. [19–21]. In fog environments, IoT devices sense the data and transmit it to the fog nodes. Some fog nodes get overloaded due to increase in data generation. As a result of this increased overload, processing time gets increased and delivery time also gets affected. The delivery time depends on the computation taking place on the fog nodes as heavy processing load on fog entities will take more delivery time [22].

To remove this issue, proper correspondence between various fog entities is of utmost importance and the overloaded entities at the fog layer should pass the load to less overloaded nodes. Resource utilization can be increased by proper load balancing among fog nodes [23]. At the fog layer, the routers are the physical servers and act as the processing entities for the services rendered by fog layer [24, 25]. The routers as the computing nodes have the capability to increase the computation as well as storage performance, which can be fully utilized. In the current era of big data [26], numerous requirements regarding performance of the IoT applications are always needed, that is why edge computing nodes are always a priority to host services for real-time applications [27, 28]. In a fog environment, the users can use computation resources, storage resources, and network in the same manner as customers use cloud resources [29]. The virtualization technique is also employed to provide resources whenever required [30].

All the applications that exploit IoT devices can be executed using fog entities and resources used in the remote cloud data center. The resources are allocated with the help of centralized and geographically distributed computing nodes for the IoT applications and the resources are scheduled after compatible fog nodes are selected to host the services rendered by fog nodes integrated with the IoT applications by employing distinct resource allocation techniques [31].

Resource allocation as well as scheduling play an important role to handle various data centers, which greatly impact on improving the resource utilization, and achieving load balancing [32–34]. In a cloud environment, workload on the physical machines (PMs) is equally distributed among all the PMs, to avoid low-latency and degraded services [35–37]. In a fog environment, resource allocation becomes more complex since computing nodes have to respond both in the fog as well as cloud environment. In a cloud environment, the computing nodes are dispersed in a centralized data center while in fog, the computing nodes are dispersed at the edge of the network [38, 39]. The resource requirements by fog nodes for the processing of IoT applications may vary as different applications possess distinct demands of computation power, bandwidth, and storage capacity [40]. Because of this, it becomes important to employ various resource allocation techniques to meet the resource requirements of applications used by end users and for the purpose of load balancing. Load balancing involves various challenges which include:

- **Efficiency**: that means the capacity of the load balancing algorithm to shift workload. It is represented using resources utilized and respective response time.
- **Overhead**: It is actually the overall burden involved in employing an algorithm for load balancing in a fog processing environment.
- **Fault-Tolerant**: It means that if failure occurs in any node, then it should not affect the whole system and load must be passed to other nodes.

1.1 *Fog Computing Architecture Layers*

Fog computing architecture uses assistance from various IoT devices such as switches, routers, multiplexers, etc. for storage and processing purposes. This architecture engages physical and logical modules of an interconnected system along with software and hardware to form an overall network of a large number of interconnecting devices. Fog node allocation (physical as well as geographical), along with the topology and protocols used, configure key architectural attributes of a fog architecture. Fog architecture engages functions at six different layers [41] as depicted in Fig. 1 and described below.

1.1.1 Physical and Virtualization Layer

- This layer incorporates several types of nodes viz. physical nodes, virtual nodes, and virtual sensor networks which are spread over different geographical locations and are managed as per their types.
- A node can be anything from a stand-alone mobile phone to a temperature sensor fitted inside a building.
- The sensors are responsible for capturing the data from their surroundings and pass the collected data to layers at the upper levels for further computing and filtering via gateways.

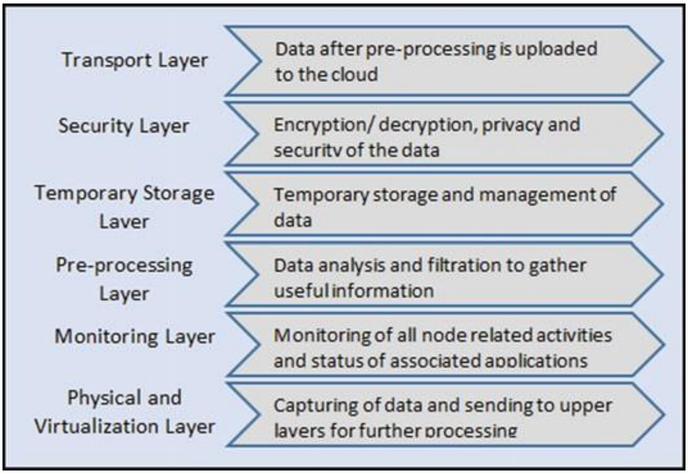


Fig. 1 Fog layer architecture

1.1.2 Monitoring Layer

- At the monitoring layer, as the name suggests, monitoring of all the activities related to various nodes is performed.
- Availability of all the nodes, sensors, network elements along with resource utilization is monitored.
- Nodes-related activities such as which node is performing what task, amount of time these nodes have worked for, what is required by nodes and at what time it is required along with additional physical attributes they own, the battery life, etc. are monitored.
- The performance and present status of all the deployed applications are also monitored.
- Energy and power consumption of fog nodes are checked from time to time.

1.1.3 Pre-processing Layer

- Pre-processing layer is accountable for information management tasks mainly related to analysis.
- Data analysis involves trimming meaningful information from mass data gathered by the end devices.
- Collected data is analyzed and filtered to extract useful information and thoroughly inspected for any errors and impurities present which are further removed.
- Data analysis is an important aspect that must be performed to gather hidden insights before data can be used for other purposes.

1.1.4 Temporary Storage Layer

- The data obtained from the pre-processing layer is then stored for a small amount of time in the temporary storage layer.
- Temporary storage layer is linked with momentary dispersion and replication of pre-processed data.
- Any tool like vSAN is employed to manage storage at the present layer.
- Data is no longer needed to store locally when it is transmitted to the cloud and may be taken out from the short-term storage layer.

1.1.5 Security Layer

- Security layer is entirely dedicated to the security and privacy of the data.
- At this layer, encryption and decryption of data occurs. Integrity procedures are also put into practice to protect data from altering.
- Privacy in case of fog computing concerns leakage of private information such as data, location, or usage.

- Privacy preservation is a challenge in fog computing which is well ensured by the security layer before the information is distributed to the fog computing entities.

1.1.6 Transport Layer

- After transformation, the information is transferred permanently to the cloud datacenters for extraction and creation of more useful services for users.
- Preferably only a portion of the data is uploaded for efficiency purposes.
- Received data is processed using smart gateways prior to sending it to the cloud layer for permanent storage.
- As there are limited fog resources, communication protocols employed need to be lightweight, efficient, and customizable.

1.2 *Fog Computing Elementary Layers*

Further, the above six-layered architecture is categorized into three elementary layers [41] as shown in Fig. 2 and the detailed description of these layers is described as below.

1.2.1 Terminal Layer (IoT Devices)

Besides, this layer is often interconnected wirelessly and composed of a large number of nodes that are deployed in that region and look like a multi-hop self-organizing sensor network. The major objective is to collect all relevant data and forward it to the upper layer for processing. The main features of this layer are:

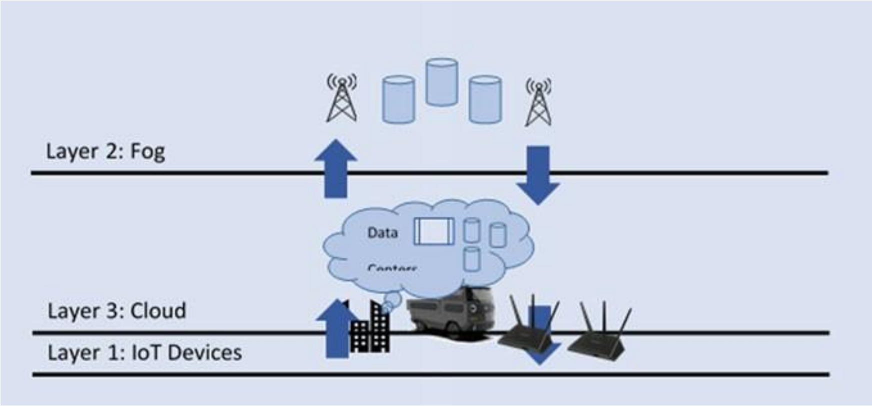


Fig. 2 Hierarchical computing architecture

- The terminal layer is the basic layer in the fog architecture and is nearest to the end clients, this layer accommodates various pieces of objects such as mobile phones, smart vehicles, sensors, smart cards, etc.
- All those objects which can function as well as compute data are a part of this layer. These are scattered over different geographic locations far away from each other.
- All these devices exhibit different platforms and architectures and mostly account for data sensing and acquiring.
- These devices can work in a varied environment, with other devices from distinct mechanisms and autonomous ways of communication.
- Along with sensing and capturing the data, these devices are also responsible for sending the data to the layer at next level.

1.2.2 Fog Layer

It is considered as a middle layer between the cloud layer and terminal layer and located at the edge of the network which is further close to the terminal layer. The main functionality of the fog layer is to perform the computing function to reduce the network transmission delay from the remote cloud. The main features of this layer are:

- Fog layer resides on the edge of a network which is very near to an end device.
- Fog layer includes numerous IoT/end devices which are referred to as fog nodes/entities such as switchers, gateways, specific fog servers, etc. These fog computing entities reside between end devices/IoT and cloud data centers [42].
- A fog entity can be a fixed geographical location such as a building, bus stop, etc. or can be something which is inside a mobile device.
- These fog entities render services to the IoT devices upon getting connected. Data that is received by the nodes is computed, transferred, and stored for a short amount of time.
- Fog nodes and cloud data centers are connected with each other through IP core networks and collaborate with cloud data centers to provide enhanced computing and storage capabilities.
- Real-time applications such as health and defense based are accomplished in the fog layer.

1.2.3 Cloud Layer

The main function of this layer is to store the data received from different fog nodes and provide a platform for communication between different availability zones for sharing of information. The main features of this layer are:

- The cloud layer resides at the utmost end of the complete fog architecture and includes numerous devices for storage and servers that have great performance capability [43].

- It performs extensive computation analysis and has permanent storage and intense processing capabilities to deal with large amounts of data for back-up and long-term access to the users.
- Huge scalable data hubs with high processing abilities renders all the fundamental facilities of cloud computing to the end users. Usually, the information stored in this layer is not required by the end users frequently.
- These data centers provide cloud resources on-demand for improved utilization and for that reason are managed with the help of some control strategies [44].

1.3 Communication Workflow in Fog Environment

In a fog domain, the IOT devices firstly communicate with fog admin by requesting tasks for execution. The fog admin forwarded the received task to the fog controller, which further communicates with the fog nodes as shown in Fig. 3. The fog controllers are basically of two types: centralized and distributed controllers. The central load balancing is served by a centralized controller that equally distributes data among all fog entities as displayed in Fig. 3. Central controller in load balancing provides convenience in the form of simpler implementation and easier management, and is faster to repair if there is some kind of loss. Central controller has information viz. resource utilization, load, and connections in a fog environment. These controllers are exercised to observe facts all around. Centralized controller collects progress of all connected nodes, performs analysis, and uses the analyzed data for resource allotment.

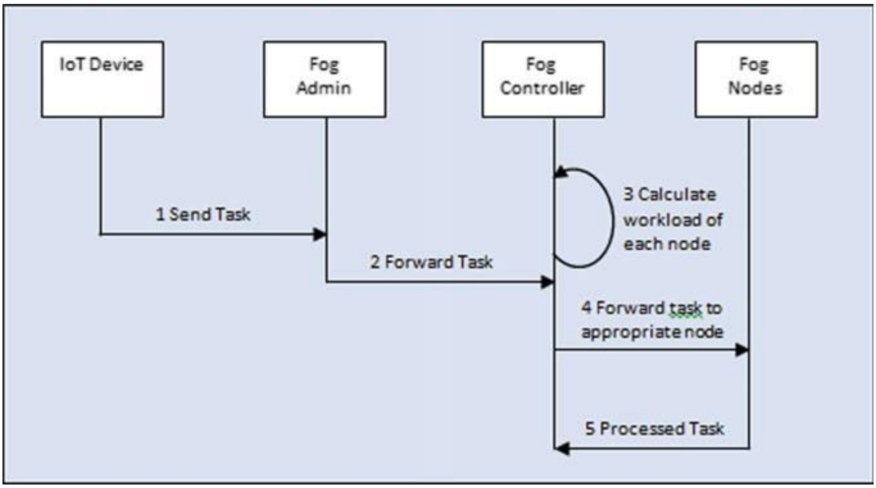


Fig. 3 Communication between FOG system

Centralized controller suffers from communication congestion because every node needs to communicate with the controller in order to perform any task. If in case of any failure at the central point it crashes, the overall system deteriorates. Also, a great amount of correspondence is needed as every node has to keep the controller busy with details all the time. To eradicate this issue, as displayed in Fig. 3, a distributed controller is employed that corresponds with the local one and the load of processing is dispersed. The distributed controller enhances the actuality as well as expandability of the network. The only pitfall with the distributed controller is that it needs liaising with the local one.

2 Related Work

Employing resources nearer to IoT/end-devices helps the system with low latency and other benefits in terms of provisioning of many new applications like mobile data offloading. This section discusses the cyber foraging, cloudlet, MEC, and fog concepts and briefly explains the background and related research.

2.1 Cyber Foraging

Cyber foraging a concept where mobile devices with minimum resources offload their work to other resource-rich devices in the surrounding region. The term 'Cyber foraging' was first coined by Satyanarayanan [45, 46] in 2001 and was further discussed in detail by Balan et al. [47] in 2002. The devices which work on behalf of mobile devices are called surrogates. These surrogates render powerful computing resources than those provided by mobile devices. Surrogates are located close to mobile devices in order to provide better response time. Now with the advancements in technology more concepts have started floating viz. cloudlet, MEC, and fog.

This concept is discussed with edge computing because of its similar processing pattern. Cyber foraging discusses [48] how the nearby devices or servers (also known as surrogates) are used for performing tasks on behalf of mobile users. These servers have high bandwidth networks and that is why they can perform computing faster than mobile devices. Since these servers are located closer to the mobile devices, therefore, the outputs are delivered faster thereby reducing latency [49, 50]. In the absence of any nearby server, the quality of service will be poor affecting the performance of mobile devices. Use of such techniques because of its reduced latency capability has increased a lot from day-to-day activities like using personal mobile handsets to areas such as military service where data is secured from unauthorized access. This technology is deployed in applications such as face detection and fingerprint sensing where data is captured locally and sent to the nearby database servers for further processing. That surrogate performs the task of comparing and gives back the results instantly.

2.2 Cloudlet

The Cloudlet concept was introduced by Satyanarayanan [46] in 2009. It is referred to as cloudlet-based cyber foraging in [48, 51] and [49, 52]. Cloudlets adopts the virtual machine (VM) based—virtualization technique for computing [53]. It is a resource-rich, trusted computer or cluster of computers that are located in the proximity of mobile devices. With the use of VMs, cloudlets perform intensive computing which is not possible by mobile devices. All real-time applications employ such techniques to provide resource-poor devices with better and reduced-latency service [51, 52]. If a nearby cloudlet is not available then mobile devices will connect to a distant cloudlet which will degrade the quality of the service [54]. Cloudlets are scalable with respect to the service requests sent by mobile devices. Due to the virtualization technique, mobile devices are able to find compatible cloudlets to offload their heavy work. Applications such as face recognition use cloudlets for face detection and matching which is performed on VMs.

Though hierarchy-wise, cloudlets fall in the middle i.e. mobile device—cloudlet—cloud. Despite being isolated from the cloud, cloudlets can function as a cloud because cloud intervention is not required in the VM provisioning of the cloudlets.

Multi-Access Edge Computing was initiated under the guidance of the European Telecommunication Standards Institute (ETSI) in the 2014 and now can be added to the architecture because of its increasing scope in the fields of video analytics, location services, augmented reality, or data caching to name a few. MEC works by processing data sent by the user and moving the computing capability closer to the end user devices. This way data doesn't have to travel longer distances; thus, it is processed faster. Applications are run locally and a real-time response rate is received. This avoids issues with latency or data congestion.

ETSI does the work of standardizing the APIs used in mobile edge platforms [55]. Few APIs have already been standardized such as radio network API [22, 51], Mobile application enablement API [41, 53], and Location API [18, 52, 56]. A survey of MEC is provided in the reference MEC is an architectural standard for edge computing which is described by ETSI as any location in a network whereas Cloudlet and fog can act as a superset to MEC [54, 57]. Fog Computing extends the cloud computing to the edge of networks. Fog nodes (FCNs) are basically located at the network edge i.e., separated from the main cloud data centers. Mobile Edge Computing (MEC) pushes the cloud computing capabilities close to the Radio Access Networks in 4G and 5G. ETSI is also developing a standard and system architecture for several APIs [58]. A cloudlet is the middle layer of a 3-tier hierarchy as: 'mobile device—cloudlet—cloud'. Goal of the cloudlet is to "bring the cloud closer to the users" [59, 60].

Fog computing has been examined from different viewpoints by various scholars and they have produced many survey papers to support their views. For example, Shi et al. [61] examined mainly latency in fog computing that plays a crucial role in various applications such as healthcare systems. Shi et al. [61] also discusses use and future prospects of fogging in healthcare systems. In addition, Yi et al. [62] surveyed and discussed fog computing, introduced various application scenarios, future challenges & opportunities, and issues that may arise during the implementation of such systems.

For secure message-passing, the study of a secure IoT is conducted by Suárez-Albela et al. [63]. They have analyzed securing gateways with minimal power consumption. In addition, by examining the key role of the Internet of Things (IoT), Bonomi et al. [64] studied the effects of integrating IoT with fog computing. Their paper also presents a hierarchy-based distributed architecture. They investigated systems like wind farms and smart traffic lights to observe the utility of the presented architecture.

Considering the importance of the Internet of Things (IoT) in the present scenario which demands low latency and better quality of service (QoS), Yousefpour et al. [65] came up with a framework to study and evaluate effects of fogging on service delays by IoT devices. Fogging provides the solution in terms of reduced latency and processing near to the end devices. A policy for minimizing the delays has also been introduced in the paper which is successfully evaluated with the help of an analytical model.

Lee et al. [66] have done a deep study and argued on the security hazards that may surface on adopting IoT fog. They have studied the current trends in IoT fog along with present-day measures that can be taken to secure IoT fog. They have also investigated the possible threats to the IoT fog.

Hong et al. [67] came up with Mobile Fog (MF), a programming model, to better study the future internet applications and performed analysis of the model using use cases to depict the effectiveness of it. They also simulated the applications to evaluate their performance.

Mahmud et al. [68] studied the present-day developments in the field of fog computing and also extensively analyzed the challenges that may occur in the fog layer. A taxonomy has been presented by them based on the study conducted. A mapping of the current developments with the taxonomy has also been done.

Cardellini et al. [69] states that storm, a DSP-based system, is extended with capabilities to perform execution of a distributed QoS-aware scheduler and made it self-adaptable. In this paper, they performed the evaluation using two different DSP applications; one with simple topology and with different requirements and the other one included applications like word count and log processing. When results were compared with that of the centralized scheduler, it was found that the QoS-aware scheduler performed better than the centralized one.

Li et al. [70] have studied the concept of fog-enhanced radio access networks (FeRANs) in their work. In case of FeRANs, to meet the demand of minimum latency, resource management is of utmost importance and it has been the topic of study of this paper. The paper also proposes two other schemes for resource management i.e. fog resource reservation & reallocation. Results show that one-hop access probability can be improved despite of heavy load in case of real-time vehicular services.

A framework by the name of INDICES has been discussed in the work of Shekhar et al. [71] that addresses performance and interference issues which earlier were not considered while adopting fog or edge technologies to reduce delay latencies. They also proved their claims with the help of results produced by a setup that consists of a centralized cloud and MDCs.

In the work of Skarlat et al. [72] the authors focused on use of all such computing devices which are present everywhere and are inexpensive to use so that IoT services can be successfully used. In their paper, they employed services of IoT devices on the fog resources while considering their QoS requirements. When results compared with the cloud-based approach, it is found that the approach used in this paper costs 35% less.

To ensure the efficient and fair utilization of services in case of private cloud providers where no money is charged for using resources, Klusácek et al. [73] discussed the issues that arise otherwise and proposed techniques for the same. They also designed a model for VM scheduler to replace the default scheduler which provides mechanisms for fair use of resources and an architecture for development of VM policies.

In order to provide faster and quality services despite heavy workloads to the end-user load is uniformly distributed among available nodes. Verma et al. [74] have come up with an algorithm that helps in balancing load in a cloud-fog environment which aims at making the environment less cloud-dependent. They have employed data replication techniques to maintain the data and have also compared the technique with load balancing techniques in cloud-based infrastructure.

In a fog environment data is offloaded to the proximal fog nodes for processing in order to provide faster services but doesn't always yield the best results because load is not equally distributed among fog nodes. Therefore, the focus of the work of Fan and Ansari, [75] is latency which is both communication latency and computing latency. They have presented a scheme to equally distribute the workload in order to reduce the latency during data flows by linking IoT devices to compatible base stations (BSs). Advantages of the presented scheme have also been verified by comparing its performance with that of other schemes through simulation techniques.

3 Proposed Framework

Due to the popularity of the cloud many efficient load balancing algorithms have been proposed so far to increase the efficiency and utilization of resources in a cloud environment. But due to the growth of IOT devices, it is not possible for the cloud to handle all incoming requests efficiently so fog computing provides the alternative. Fog computing is a new paradigm, so not much work has been carried out in this area. The fog computing environment is not a different paradigm as it provides an intermediate layer between the client layer and cloud layer, to reduce the workload on the cloud and also make the execution of tasks efficiently with a limited number of resources. As the execution of the request is performed at the fog layer so it can work with a limited number of resources i.e., less bandwidth, cost, and time as the processing is pushed closer to the end clients.

Fog computing also has data centers as cloud, so to improve the utilization of shared resources an appropriate load balancing mechanism is desirable [44]. In this proposed work a fog computing framework has been proposed that divides the virtual machines into groups named cluster and each cluster having virtual machine manager which is responsible for the local scheduling and monitoring of virtual machines as shown in Fig. 4. Further, each VM manager communicates with the cluster master server and updates the status of virtual machines periodically i.e., load on VMs, availability of VMs, etc. Whenever the client requests for task execution, it sends the request to the cluster master server (CMS) which further communicates with all VM managers and allocates the VMs based upon PSO-based approach named as Optimized Load Balancing Algorithm (OLBA) as shown in Fig. 4. The steps of the proposed frameworks are as follows:

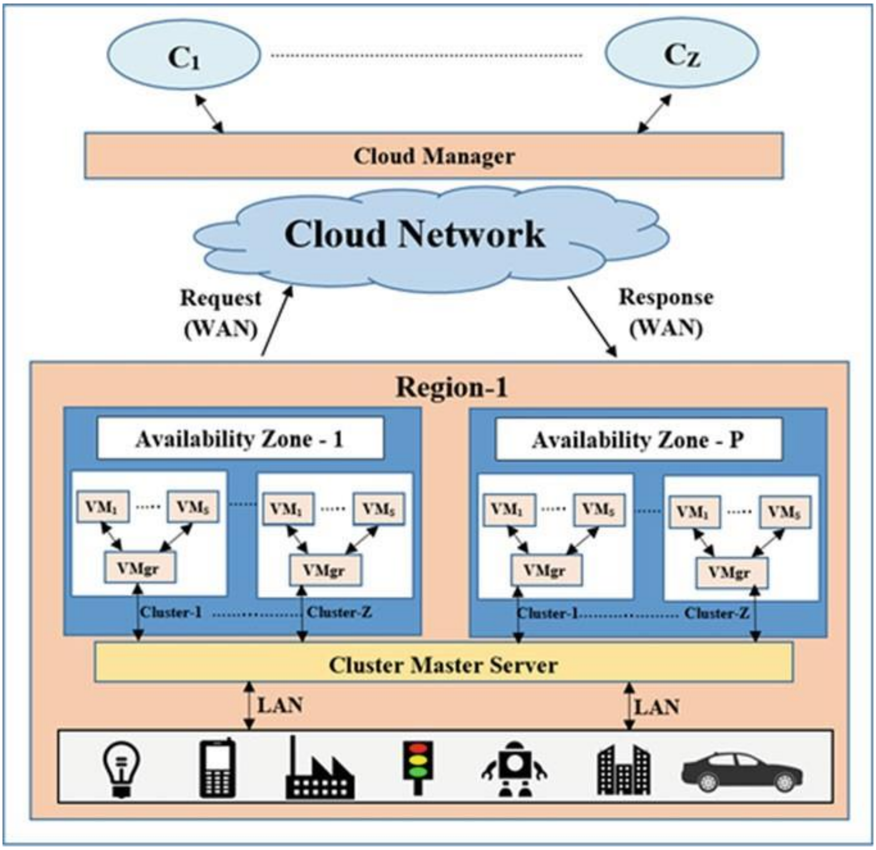


Fig. 4 Proposed framework

Step 1. Cloud and fog computing layers consist of data centers that comprise clusters and each cluster, in turn, comprises numerous VMs and one VM manager.
Step 2. The fog computing layer consists of a Cluster Master Server (CMS) which communicates with the VM manager of every cluster.
Step 3. Initially, when an IoT device or a client sends its request, it will be forwarded to the nearest Cluster Master Server of its region.
Step 4. Cluster Master Server will allocate a virtual machine in the region with the help of the proposed algorithm i.e., Optimized Load Balancing Algorithm (OLBA) which is based on the PSO approach. OLBA procedure is explained in detail in the next section.
Step 5. Initially, when an IoT device or a client sends its request, it will be forwarded to the nearest Cluster Master Server of its region.
Step 6. Cluster Master Server will allocate a virtual machine in the region with the help of the proposed algorithm i.e., Optimized Load Balancing Algorithm (OLBA) which is based on the PSO approach. OLBA procedure is explained in detail in the next section.
Step 7. If the requirement for VM increases during the execution of real tasks then Cluster Master Server will communicate with the nearby cluster master servers in different availability zones in the same region so that execution of real tasks becomes feasible.
Step 8. If during the execution of Step 5, the required data is not available within the same region, then the cluster master will forward the request to the cloud server.
Step 9. Finally, data will be sent to the cloud for processing and will be available for further communication.

3.1 Optimized Load Balancing Algorithm (OLBA)

As discussed above, Cluster Master Server will allocate a virtual machine to the requesting client in the region with the help of the proposed approach i.e., OLBA. This approach is based on Particle Swarm Optimization (PSO) technique. This is a computational method of measuring the quality by iteratively optimizing a problem in local spaces to find the local best and then to compare all the local best to find the ultimate global best solution. Here all the IoT devices are working at the last layer and are communicating with the cluster master server (CMS) as shown in the proposed framework in Fig. 4. As defined above all virtual machines of a region are grouped into different equal size clusters. For every new request cluster master server (CMS) looks for the best available in terms of efficiency and load. The VM from every cluster of the region as the local best solution. Henceforth, all selected VMs from different clusters participate in the selection of the globally best VM to handle the request. This selection is repeated for all requests received from any of

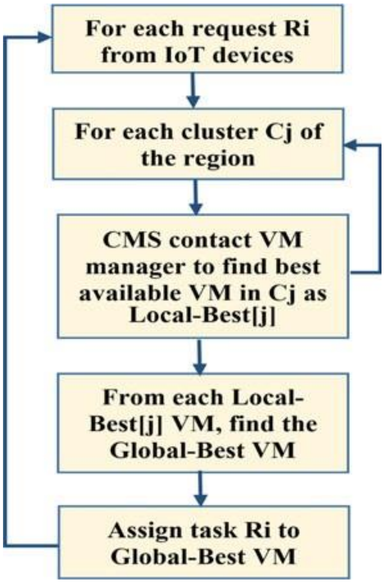


Fig. 5 Working steps of proposed framework

the IoT devices. The detailed working of the algorithm OLBA is drawn in the form of the flow chart in the Fig. 5 and the steps of the algorithm are as follows:

1. Initialize the N number of fog devices $F_1, F_2 \ldots F_N$.
2. Estimate the M number of incoming requests $R_1, R_2 \ldots R_M$.
3. Estimate total of Z clusters as $C_1, C_2 \ldots C_Z$.
4. Allocate the fog devices to each cluster (C_i) and the size of cluster (CS) is calculated as:

$$\text{Size of cluster} = \text{Total no. of fog devices/No. of clusters}$$

$$S = N/M$$

5. Assign each cluster with S number of VMs as $VM_1, VM_2 \ldots, VM_S$.
6. For each incoming request $R_i = R_1, R_2 \ldots R_M$ do:

   (a) For each cluster $C_j \in_1, C_2 \ldots C_Z$, do
      i. Find out the local best virtual machine having better efficiency (MIPS) and least loaded in C_j
      ii. Store the index of the best virtual machine of C_j in the array, Local-Best[j]
      // End of For Loop of step (a)

   (b) For each cluster indexed $j \neq \ldots Z$, do
      i. Find out the global best machine for R_i having better efficiency (MIPS) and least loaded among the local best machines selected in each cluster, from the array: Local-Best[j].
      // End of For Loop of step (b)

(c) Assign the task R_i to the global best machine.
(d) Repeat step 5 until all requests/tasks have been completed.
// End of For Loop of step 5.

In the above OLBA algorithm, firstly fog devices are initialized in the first step. In the second step, a number of incoming requests from different user's are considered. After that, a number of clusters are identified and compute the size of the cluster by dividing the number of clusters in third and fourth steps. Every cluster is initialized with a number of virtual machines according to their size. For every incoming request, compute the best local resource which has the least load in each cluster in step 6(a). After that global optimization is done. A Comparison between the best least loaded machine from each cluster is performed at global level. Assign the request to the best virtual machine at global level in step 6(b). As a result, the task is always assigned to the most appropriate and optimized selected virtual machine.

For clear understanding of the above-described procedure, the detailed steps are also explained in the form of the activity diagram shown in Fig. 5.

3.2 Environmental Setup

In order to test and compare the experimental results, the simulations have been carried out in a cloud-fog environment using iFogSim to measure the efficiency of our proposed framework [75]. The iFogSim is an open-source platform with an extension of cloud sim simulator. iFogSim uses the simulation/modeling of computing resources in fog environments to test the efficiency of managing the resources with the help of different scheduling scenarios and policies The homogenous characteristics of computations is considered only for simulation scenarios. It considers different tasks with varying size that are submitted on VM's having different parameters. Due to the random environment, the simulations received various types of requests, i.e., memory, data, data transfer operation, etc. The fog nodes in the framework having their RAM, CPU, utilization cost, and transmission capacity along with processing speed, i.e., MIPS. During simulation setup, 15 processing nodes were selected, with the specifications as shown in Table 1.

The major responsibility of the proposed framework is to cater to all requests received from the IoT devices. Each request is distributed into various task tuples and they are further assessed and divided based upon the required processing. The task submitted for execution has various requirements i.e., the memory required, data transfer capacity, and input/output which is presented in Table 2 (Table 3).

Table 1 Simulation parameters for fog environment

Parameters	Cloud	Fog
Number of nodes	5	10
Usage cost of CPU ($)	2.0	0.4

Table 2 Properties for each incoming request

Property	Value
Number of instructions (MIPS)	[1–200]
Required memory (MB)	[50–200]

Table 3 Simulation setup characteristics

Property	Value
System	Intel Core i5, 2 GHz
Operating system	Windows 10

The proposed framework has been tested using between 20 and 400 tasks. The simulation environment for the proposed framework is shown in Table 1. The Java with Eclipse and iFogSim is chosen for simulation because iFogSim is reliant on CloudSim, which is the most popular and widely accepted tool for conducting simulation in the cloud computing environment [76]. The proposed approach topology is demonstrated in Fig. 6, which also has been simulated using iFogSim to check the efficiency of the proposed system.

Results of our proposed framework are compared with existing scheduling algorithms such as first come first served (FCFS) and shortest job first (SJF), Max_Min. These scheduling algorithms are selected for comparison as these are the standard scheduling algorithms. In FCFS, resources are allocated to tasks according to the arrival time of tasks and if the resources are busy then tasks have to wait till the availability of resources whereas using SJF resources are allocated to tasks according to their size (execution time). In Max_Min Task have been selected based upon their completion time. In the proposed work the tasks are allocated to the most appropriate resources and balance the load as tasks are submitted to the most appropriate resource, not in a random way i.e., resources are assigned based on the requirements of the incoming requests.

The proposed algorithm treats equally all fog nodes and provides the best resource utilization of all fog nodes. A small number of resources are used by traditional algorithms and do not provide resource utilization thus load is increasing.

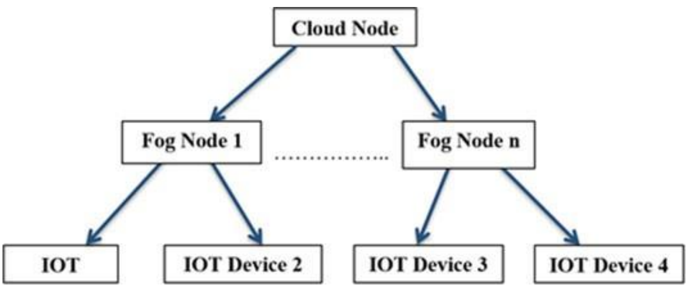


Fig. 6 Proposed framework topology designed in iFogSim

4 Results and Discussions

Load balancing depends on various parameters such as turnaround time, resource utilization, delay, response time and can be improved by managing these parameters carefully [23, 76]. To assess and compare various load balancing mechanisms, several metrics are required to measure qualitative paradigms [77].

The simulation results of the proposed approach are analyzed on the parameters of: resource utilization, turnaround time, delay, average response time [78]. Further, proposed approach is compared with existing popular traditional scheduling algorithms i.e. first come first served (FCFS) and shortest job first (SJF), Max_Min. In the next subsection, the turnaround time comparison is performed. Also, in further subsections utilization of resources are tested. Section 3 measures the response time of nodes. Finally, subsection IV tested the delay comparison of nodes**.**

4.1 Turn Around Time Performance Test

It can be defined as the time of interval between submission and completion of tasks in a fog environment. The proposed algorithm is compared with the FCFS, Max_Min, SJF algorithms, and simulation work carried out using 10 different workloads ranging from 40 to 400 tasks with size ranging from 2000 to 8000. All tasks are executed on the VMs with computation speed 2000 MIPS each. The obtained result compared with traditional algorithms as shown in Fig. 7.

The results show that the turnaround time in FCFS is rapidly increased due to its non-preemptive properties, as the short tasks have to wait more due to long tasks.

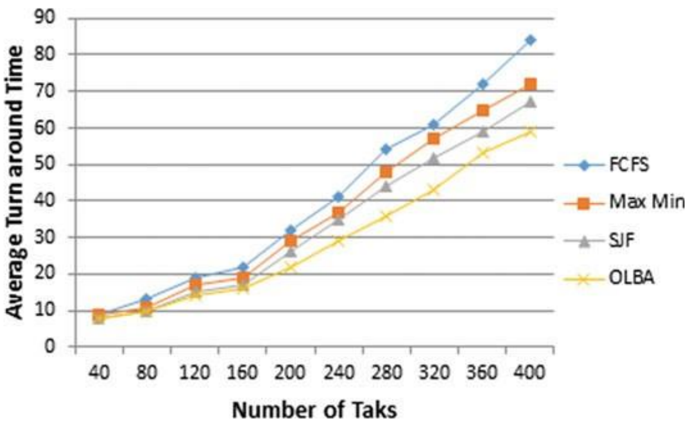


Fig. 7 Turnaround time test comparison

Similarly, Max–Min algorithm also has a long turnaround time because it assigns the longest task first to VMs having minimum remaining execution time. SJF shortest jobs are served first and delay is lesser than FSFC and Min–Max but there is no appropriate selection of the resources as proposed by OLBA.

4.2 Fog Resources Resource Utilization

Resource utilization is the most important factor in load balancing. It can be calculated as the subtraction of starting execution time from finishing execution time of tasks for each resource.

$$Ru_j = \Sigma t_i \text{ where } t_i \text{ will be executed on } R_j \dot{\ } t_{fi} - t_{si}{}^{\Sigma} \tag{1}$$

where, t_{fi} is the finishing time and t_{si} is the start time of task t_i on resource R_j.

The proposed algorithms are compared with FCFS, Max_Min, and SJF in terms of resource utilization (in percentage). In Fig. 8 shows that using traditional approaches, there is variation in resource utilization. Since in FCFS resources are allocated based on the order of requests arrived. Thus, resource utilization is not uniform. Whereas the proposed algorithm finds the most appropriate fit using local best and global best optimization techniques. Henceforth, resource utilization is uniformly divided among all the fog nodes.

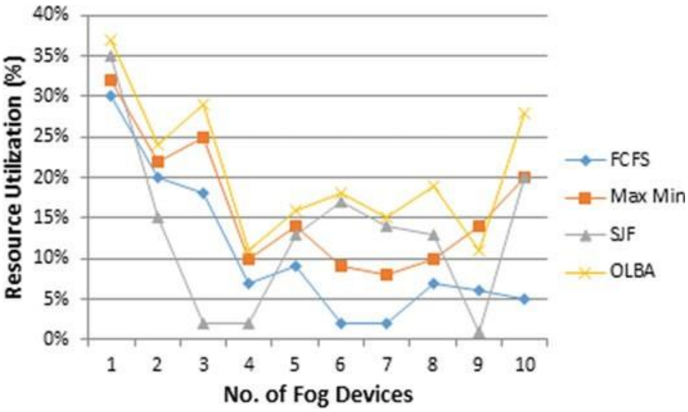


Fig. 8 Utilization of resources in terms of tasks performed by fog device

4.3 Average Response Time

It describes the response time of a server in the fog environment as the duration of the time interval starting from the submission of a request to sending the response for the same [37]. Equation 2 presents the computation of response time.

$$R_t = T_p + T_c + T_d \tag{2}$$

where, R_t represents the response time, T_p is the propagation time, T_c shows computation time and T_d presents the delay time.

Figure 9, shows the simulation results with 60 incoming tasks in a cluster of 10 and 20 fog devices, it has been shown that the OLBA provides reasonably better results in terms of average response time in comparison with traditional approaches like FCFS, SJF, Max_Min.

4.4 Processing Delay

The fog node processing delay is the time that a node spends processing a packet. In cloud environment resources are remotely deployed which increase the latency inflow of tasks submitted in cloud but in fog computing resources are near to IOT devices henceforth delay is significantly reduced [72]. It can be computed using following formula:

$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}} \tag{3}$$

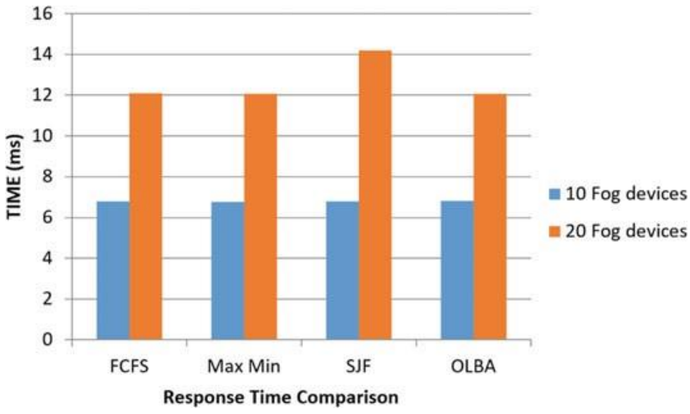


Fig. 9 Average response time comparison of FCFS, Max_Min, SJF, and OLBA

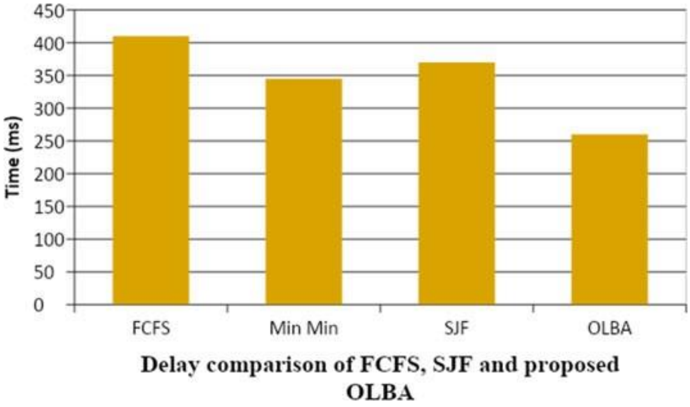


Fig. 10 Delay comparison of FCFS, Max_Min, SJF and OLBA

where d_{proc} is the processing time, d_{queue} is the delay time in queue d_{trans} transmission delay time, d_{prop} represents the propagation delay.

Figure 10, demonstrates the fog servers' processing delay after applying different task scheduling policies as it serves several requests. The x-axis and y-axis represent the different task scheduling policies and processing time in milliseconds respectively.

The overall processing delay is directly related to the strategy used for allocating the best available resource for every incoming request. If the tasks are not allocated to the appropriate fog resources then it will decrease the resource utilization and affect the overall performance of the network due to the higher value of processing delay. Henceforth, the processing delay factor is minimized to enhance the performance of the system, i.e., allocate the resources dynamically and intelligently will reduce the processing delay factor at the server end. Hence, OLBA outperforms traditional scheduling policies in terms of the overall average processing delay considered in our simulations.

Hence the simulation results prove that the above discussed proposed work has improved results in the case of turnaround time, average response time, resource utilization, and processing delay in comparison to other traditional algorithms.

5 Conclusion and Future Scope

Fog computing paradigm is popular because it enhances the cloud computing performance as the fog resources are placed near the IOT devices which reduces the latency overheads. The work presents a Load Balancing Algorithm thoroughly in Fog computing environments. An adaptive PSO-based Optimized Load Balancing Algorithm (OLBA) framework, has been proposed to balance the load between fog devices and improve QoS parameters i.e., resource utilization, response time, and delay parameters of fog devices. The results have been evaluated and verified thoroughly using simulations. The simulation results present that the proposed framework improves resource utilization, reduces the turnaround time, delay and average response time when compared with existing popular traditional scheduling algorithms, and also improves the load balancing. Present work will be enhanced by using intelligent learning-based techniques, i.e., reinforcement learning to manage resources in a fog environment and also include other QoS parameters such as security, reliability, etc. Further proposed algorithm OLBA can be enhanced in terms of performance by designing artificial intelligence and neural-based models for low priority tasks.

References

1. Vaquero LM, Rodero-Merino L, Caceres J, Lindner M et al (2008) A break in the clouds: towards a cloud definition. SIGCOMM Comput Commun Rev 39(1):50–55
2. Zhang Q, Cheng L, Boutaba R et al (2010) Cloud computing: state-of-the-art and research challenges. J Internet Serv Appl 1(1):7–18
3. Rajeev T, Neeraj K (2012) A novel hybrid approach for web caching. In: 2012 sixth international conference on innovative mobile and internet services in ubiquitous computing, IEEE
4. Harnal S, Chauhan RK (2016) Multimedia support from cloud computing: a review. In: 2016 international conference on microelectronics, computing and communications (MicroCom). IEEE, pp 1–6
5. Sharma G, Harnal S, Miglani N, Khurana S et al (2021) Reliability in underwater wireless sensor networks, in energy-efficient underwater wireless communications and networking, IGI Global. In: Advances in intelligent systems and computing, vol 468. Springer. https://doi.org/10.1007/978-981-10-1675-2_58
6. Jiao L, Friedman R, Fu X, Secci S, Smoreda Z, Tschofenig H et al (2013) Cloud-based computation offloading for mobile devices: state of the art, challenges and opportunities. In: 2013 future network mobile summit, pp 1–11
7. Stojmenovic I (2014) Fog computing: a cloud to the ground support for smart things and machine-to-machine networks. In: Telecommunication networks and applications conference (ATNAC), 2014 Australasian, pp 117–122
8. Yangui S, Ravindran P, Bibani O, Glitho RH, Hadj-Alouane NB, Morrow MJ, Polakos PA et al (2016) A platform as-a-service for hybrid cloud/fog environments. In; 2016 IEEE international symposium on local and metropolitan area networks (LANMAN), pp 1–7
9. Zhu X, Chan DS, Hao Hu, Prabhu MS, Ganesan E, Bonomi F et al (2015) Improving video performance with edge servers in the fog computing architecture. Intel Technol J 19(1):202–224
10. Yangui S, Tata S (2015) The SPD approach to deploy service-based applications in thecloud. Concurr Comput Pr. Exper 27(15):3943–3960
11. Pop D, Iuhasz G, Craciun C, Panica S et al (2016) Support services for applications execution in multi-clouds environments. In: 2016 IEEE international conference on autonomic computing (ICAC), pp 343–348
12. Martino BD (2014) Applications portability and services interoperability among multiple clouds. IEEE Cloud Comput 1(1):74–77
13. Massonet P, Naqvi S, Ponsard C, Latanicki J, Rochwerger B, Villari M et al (2011) A monitoring and audit logging architecture for data location compliance in federated cloud infrastructures. In: 2011 IEEE international symposium on parallel and distributed processing workshops and PhD forum (IPDPSW), pp 1510–1517

14. Bonomi F, Milito R, Zhu J, Addepalli S et al (2012) Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on mobile cloud computing. New York, pp 13–16
15. OpenFog Consortium (2017) OpenFog reference architecture for fog computing. Technical report
16. Rajeev T (2019) Automated parking system-cloud and IoT based technique. Int J Eng Adv Technol (IJEAT) 8.4 C:116–123
17. Bonomi F, Milito R, Natarajan P, Zhu J et al (2014) Fog computing: a platform for internet of things and analytics. In: Bessis N, Dobre C (eds) Big data and internet of things: a roadmap for smart environments. Springer International Publishing, pp 169–186
18. Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are (2015) Cisco White Paper. Available online: https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf. Accessed 25 Nov 2018
19. Lyu X, Tian H, Ni W, Zhang Y, Zhang P, Liu RP et al (2018) Energy-efficient admission of delay-sensitive tasks for mobile edge computing. IEEE Trans Commun 66(6):2603–2616
20. Lyu X, Ni W, Tian H, Liu RP, Wang X, Giannakis GB, Paulraj A et al (2017) Optimal schedule of mobile edge computing for internet of things using partial information. IEEE J Sel Areas Commun 35(11):2606–2615
21. Lyu X, Ren C, Ni W, Tian H, Liu RP et al (2018) Distributed optimization of collaborative regions in large-scale inhomogeneous fog computing. IEEE J Sel Areas Commun 36(3):574–586
22. Ashish C, Kumar RN (2019) A review of load balancing in fog computing, international conference on information technology (ICIT)
23. Sharma G, Harnal S, Miglani N, Khurana S et al (2020) Reliability of cluster head node based upon parametric constraints in underwater acoustic sensor networks. Underwater Technol
24. Wang S, Lei T, Zhang L, Hsu C-H, Yang F et al (2016) Offloading mobile data traffic for QoS-aware service provision in vehicular cyber-physical systems. Futur Gener Comput Syst 61:118–127
25. Yi S, Li C, Li Q et al (2015) A survey of fog computing: concepts, applications and issues. In: Proceedings of the workshop on mobile big data (Mobidata '15). ACM, Hangzhou, pp 37–42
26. Ishan S (2018) Analysis of mahout big data clustering algorithms. In: Intelligent communication, control and devices. Springer, pp 999–1008
27. Xu XL, Zhao X, Ruan F et al (2017) Data placement for privacy-aware applications over big data in hybrid clouds. In: Security and communication networks, vol 2017, Article ID 2376484, 15 pages
28. Xu X, Dou W, Zhang X, Hu C, Chen J et al (2017) A traffic hotline discovery method over cloud of things using big taxi GPS data. In: Software: practice and experience, vol 47, no 3, pp 361–377
29. Aazam M, Huh E-N (2014) Fog computing and smart gateway based communication for cloud of things, pp 464–470
30. Bonomi F, Milito R, Natarajan P, Zhu J et al (2014) Fog computing: a platform for internet of things and analytics. In: Big data and internet of things: a roadmap for smart environments, vol 546 of studies in computational intelligence, pp 169–186
31. Srivastava R, Sandeep K, Rajeev T, et al eds (2020) Green information and communication systems for a sustainable future. CRC Press
32. Xu X, Zhang X, Khan M, Dou W, Xue S, Yu S et al (2017) A balanced virtual machine scheduling method for energy-performance trade-offs in cyber-physical cloud systems. Future Gen Comput Syst
33. Yu L, Chen L, Cai Z, Shen H, Liang Y, Pan Y et al (2016) Stochastic load balancing for virtual resource management in datacenters. IEEE Trans Cloud Comput 1–14
34. Sahu Y, Pateriya RK, Gupta RK et al (2013) Cloud server optimization with load balancing and green computing techniques using dynamic compare and balance algorithm. In: Proceedings of the 5th international conference on computational intelligence and communication networks, CICN 2013, pp 527–531

35. Soni G, Kalra M (2014) A novel approach for load balancing in cloud data center. In: Proceedings of the 2014 4th IEEE international advance computing conference, IACC 2014, pp 807–812
36. Xu X, Dou W, Zhang X, Chen J et al (2016) EnReal: an energy-aware resource allocation method for scientific workflow executions in cloud environment. IEEE Trans Cloud Comput 4(2):166–179
37. Li S, Zhang Y (2016) On-line scheduling on parallel machines to minimize the makespan. J Syst Sci Complexity 29(2):472–477
38. About open edge computing initiative. [Online]. Available: http://openedgecomputing.org/about.html. Accessed on 13 Sept 2017
39. Mach P, Becvar Z (2017) Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun Surv Tutor 19(3):1628–1656
40. Sumit K, Rajeev T (2021) Dynamic popularity window and distance-based efficient caching for fast content delivery applications in CCN. Eng Sci Technol Int J
41. https://www.educba.com/fog-computing-architecture/
42. Miglani N, Sharma G (2019) Modified particle swarm optimization based upon task categorization in cloud environment. Int J Eng Adv Technol (IJEAT)
43. Miglani N, Sharma G (2018) An adaptive load balancing algorithm using categorization of tasks on a virtual machine based upon queuing policy in cloud environment. Int J Grid Distrib Comput
44. Harnal S, Chauhan RK (2020) Towards secure, flexible and efficient role based hospital's cloud management system: case study. EAI Endorsed Trans Pervasive Health Technol 6(23)
45. Satyanarayanan M (2001) Pervasive computing: vision and challenges. IEEE Pers Commun 8(4):10–17
46. Satyanarayanan M, Bahl P, Caceres R, Davies N et al (2009) The case for VM-based cloudlets in mobile computing. IEEE Pervasive Comput 8(4):14–23
47. Balan R, Flinn J, Satyanarayanan M, Sinnamohideen S, Yang H-I et al (2002) The case for cyber foraging. In: Proceedings of the 10th workshop on ACM SIGOPS European workshop, New York, pp 87–92
48. Lewis GA, Echeverría S, Simanta S, Bradshaw B, Root J et al (2014) Cloudlet-based cyber-foraging for mobile systems in resource-constrained edge environments. In: Companion proceedings of the 36th international conference on software engineering. New York, pp 412–415
49. Ali AMM, Ahmad NM, Amin AHM et al (2014) Cloudlet-based cyber foraging framework for distributed video surveillance provisioning. In: 2014 fourth world congress on information and communication technologies (WICT), pp 199–204
50. Hu YC, Patel M, Sabella D, Sprecher N, Young V et al (2015) ETSI white paper N011. In: Mobile edge computing: a key technology towards 5G, 1st ed
51. ETSI GS MEC 11 (2017) Mobile edge computing (MEC): mobile edge platform application enablement
52. ETSI GS MEC 12 (2017) Mobile edge computing (MEC): radio network information API
53. ETSI GS MEC 003 (2016) Mobile edge computing (MEC): framework and reference architecture
54. Taleb T, Samdanis K, Mada B, Flinck H, Dutta S, Sabella D et al (2017) On multi-access edge computing: a survey of the emerging 5g network edge cloud architecture and orchestration. IEEE Commun Surv Tutor 19(3):1657–1681
55. About Open Cord Initiative. [Online]. Available: http://opencord.org/about. Accessed on 13 Sept 2017
56. Smallen S, Casanova H, Berman F et al (2001) Applying scheduling and tuning to on-line parallel tomography. In: Proceedings of the 2001 ACM/IEEE conference on supercomputing. Denver, CO
57. Stiles JR, Bartol TM, Salpeter EE, Salpeter M et al (1998) Monte Carlo simulation of neuro-transmitter release using MCell, a general simulator of cellular physiological processes. In: Bower JM (ed) Computational neuroscience. Springer, Boston, MA

58. Chiang M, Zhang T (2016) Fog and IoT: an overview of research opportunities. IEEE Internet Things J 3:854–864 [CrossRef]
59. Bitam S, Zeadally S, Mellouk A et al (2017) Fog computing job scheduling optimization based on bees swarm. Enterp Inf Syst 12:373–397 [CrossRef]
60. Peter N (2015) Fog computing and its real time applications. Int J Emerg Technol Adv Eng 5:266–269
61. Shi Y, Ding G, Wang H, Roman HE, Lu S et al (2015) The fog computing service for healthcare. In: Proceedings of the 2015 2nd international symposium on future information and communication technologies for ubiquitous healthcare (Ubi-HealthTech). IEEE, Beijing, pp 1–5
62. Yi S, Li C, Li Q et al (2015) A survey of fog computing: concepts, applications and issues. In: Proceedings of the 2015 workshop on mobile big data. Santa Clara, CA, pp 37–42
63. Suárez-Albela M, Fernández-Caramés T, Fraga-Lamas P, Castedo L et al (2017) A practical evaluation of a high-security energy-efficient gateway for IoT fog computing applications. Sensors, 17 [CrossRef] [PubMed]
64. Bonomi F, Milito R, Natarajan P, Zhu J et al (2014) Fog computing: a platform for internet of things and analytics. In: Big data and internet of things: a roadmap for smart environments. Springer, Berlin, pp 169–186
65. Yousefpour A, Ishigaki G, Jue JP et al (2017) Fog computing: towards minimizing delay in the internet of things. In: Proceedings of the 2017 IEEE international conference on edge computing (EDGE). Honolulu, HI, pp 17–24
66. Lee K, Kim D, Ha D, Rajput U, Oh H et al (2015) On security and privacy issues of fog computing supported internet of things environment. In: Proceedings of the 2015 6th international conference on the network of the future (NOF). Montreal, QC, pp 1–3
67. Hong K, Lillethun D, Ramachandran U, Ottenwälder B, Koldehofe B et al (2013) Mobile fog: a programming model for large-scale applications on the internet of things. In: Proceedings of the second ACM SIGCOMM workshop on mobile cloud computing. Hong Kong, pp 15–20
68. Mahmud R, Kotagiri R, Buyya R et al (2018) Fog computing: a taxonomy, survey and future directions. In: Internet of everything. Springer, Singapore, pp 103–130
69. Cardellini V, Grassi V, Presti FL, Nardelli M et al (2015) On QoS-aware scheduling of data stream applications over fog computing infrastructures. In: Paper presented at: 2015 IEEE symposium on computers and communication (ISCC), Larnaca, Cyprus
70. Li J, Natalino C, Van DP, Wosinska L, Chen J et al (2017) Resource management in fog-enhanced radio access network to support real-time vehicular services. In: Paper presented at: 2017 IEEE 1st international conference on fog and edge computing (ICFEC). Madrid, Spain
71. Shekhar S, Chhokra AD, Bhattacharjee A, Aupy G, Gokhale A et al (2017) INDICES: exploiting edge resources for performance-aware cloud-hosted services. In: Paper presented at 2017 IEEE 1st international conference on fog and edge computing (ICFEC). Madrid, Spain
72. Skarlat O, Nardelli M, Schulte S, Dustdar S et al (2017) Towards QoS-aware fog service placement. In: Paper presented at 2017 IEEE 1st international conference on fog and edge computing (ICFEC), Madrid, Spain
73. Klusácek D, Parák B, Podolníková G, Ürge A et al (2017) Scheduling scientific workloads in private cloud: problems and approaches. In: Proceedings of the 10th international conference on utility and cloud computing (UCC). Austin, TX
74. Verma S, Yadav AK, Motwani D, Raw R, Singh HK et al (2016) An efficient data replication and load balancing technique for fog computing environment. In: Paper presented at 2016 3rd international conference on computing for sustainable global development (INDIACom). New Delhi
75. Fan Q, Ansari N (2018) Towards workload balancing in fog computing empowered IoT. IEEE Trans Netw Sci Eng. Early access
76. Xu X, Fu S, Cai Q, Tian W, Liu W, Dou W, Sun X, Liu AX et al (2018) Dynamic resource allocation for load balancing in fog environment. Wirel Commun Mob Comput, Article ID 6421607, 15 pages. https://doi.org/10.1155/2018/6421607

77. Manisha V, Neelam B, Kumar YA et al (2016) Real time efficient scheduling algorithm for load balancing in fog computing environment. Int J Inf Technol Comput Sci (IJITCS) 8(4):1–10. https://doi.org/10.5815/ijitcs.2016.04.01
78. Rajeev T, Neeraj K (2015) Minimizing query delay using co-operation in ivanet. Proc Comput Sci 57:84–90

Fog Computing in Industry 4.0: Applications and Challenges—A Research Roadmap

Sita Rani, Aman Kataria, and Meetali Chauhan

Abstract Expeditious technical developments have remodeled the industrial sector. These developments vary from mechanization of industrial tasks to autonomous industrial processes in which no human intervention is needed for regular working. An advanced concept i.e. Industrial Internet of Things (IIoT) evolved with the appliance of Internet of Things (IoT) in industrial processes; gave a new dimension to the technological advancements in the industrial sector by facilitating industrial processes with the support of Internet. Impeding the interpretation of IIoT to the production process supported another sub-domain of IoT, recognized as Industry 4.0. The concept of Industry 4.0 is realized using sensor networks, automated business processes, robots, smart equipment and machines, actuators, and people. Consequently, a huge volume of disparate data is initialized for analysis and processing. In industry, most of the processes are real-time. To avoid communication delays and ensure data security, the majority of the processes are completed locally and only necessary data is transferred over the Internet for cloud storage. To fulfill this objective, there is always a high requirement of a middleware amidst industrial processes/tools and cloud. In this connection, Fog is the most workable solution for distinct industrial scenarios. In the manufacturing industry, it can facilitate local processing along with tolerable communication delay to robots and actuators. Data gathered from various industrial processes is usually disorganized which needs preprocessing for refinement using Fog locally then communicated to the cloud. So, fog computing plays a vital role in various Industry 4.0 applications by resolving various issues. But the deployment of Fog computing in Industry 4.0 also faces a lot many challenges of different kinds related to programmability, security, heterogeneity, and interoperability. In this book chapter, we present an overview of Fog computing along with the architectural framework of Industry 4.0. We discussed the various applications of Fog computing in industry 4.0 in detail. Different problems faced in the implementation of fog computing in Industry 4.0 will be discussed. We

S. Rani M. Chauhan
Department of Computer Science and Engineering, Gulzar Group of Institutes, Khanna-141401, Punjab, India

A. Kataria (✉)
CSIR-CSIO, Chandigarh-160030, India

R. Tiwari et al. (eds.), *Energy Conservation Solutions for Fog-Edge Computing Paradigms*, Lecture Notes on Data Engineering and Communications Technologies 74,
https://doi.org/10.1007/978-981-16-3448-2_9

have also introduced various research challenges to be dealt with for the efficient deployment of fog in Industry 4.0.

Keywords Fog · Industry 4.0 · Internet of Things (IoT) · Industrial Internet of Things (IIoT) · Internet of Vehicles (IoV)

1 Introduction

The advancement in technology has brought a big revolution in the industrial sector. This advancement has a wide range of applications from automation to autonomous processes where human intervention is least required. This technological revolution is known as Industrial Internet of Things (IIOT). Industry 4.0 comprises of robots, sensor networks, actuators, business processes and applications. Nowadays, various industrial processes need to be secure and also require data to be communicated using internet, web services and cloud. For proper management of data between industrial processes and cloud/web services, there is a requirement of middleware support. The Fog is that middleware, which facilitates various industrial tasks for data processing, storage, and management in IIOT [1].

Industrial revolution started with manufacturing systems in the domains of electrical, electronic, computer, and mechanical to Industry 4.0 [2, 3]. But it requires more technological and innovative infrastructure for responding back to customer needs [4]. Industry 4.0 is a trending concept which has brought together existing technologies for making industries efficient, distributed, intelligent, and secure. On the other side, the integration of IOT with other technologies has led to threats in terms of computing, transmission, and storage. The components of industry 4.0 comprises Cyber Physical Systems (CPS), Internet of Services (IOS), Big data, Data Mining, cloud computing, and IOT [5, 6].

Besides the specific cloud version that responds to industry, cloud manufacturing (CM) lacks efficiency to manage complex information and distributed devices. CISCO launched a new approach to enhance the performance of cloud computing. This new approach is termed as the “Fog computing”. Fog computing has shown its efficiency for large IOT supported applications in various fields such as healthcare, smart cities, transportation, security, traffic control, automation, etc. The term “Industry 4.0” was referenced to strengthen manufacturing sector in the beginning [1]. Basically, The Industry 4.0 is a combination of two main technologies i.e. CPS and IOT [7].

1.1 Fog Computing

Fog computing is an extension of cloud computing [8], which has enabled most of the industrial applications as well as services to relate with IOT [9]. Fog computing

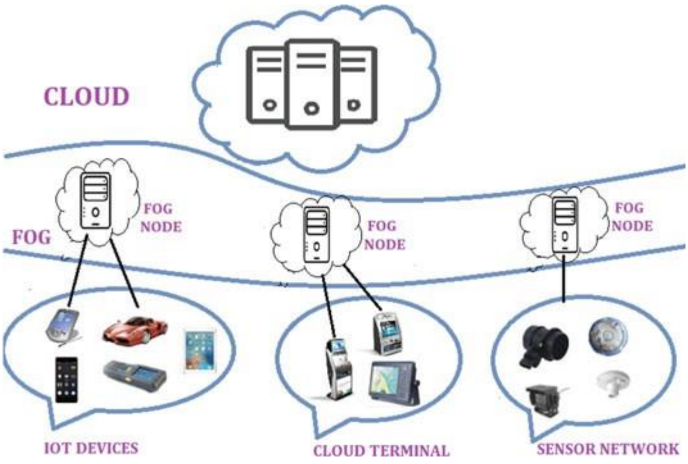


Fig. 1 Fog nodes

is a kind of scenario where wireless and decentralized devices communicate with networks to perform processing and storage tasks without interference of any third party in the industrial applications [10]. Fog computing consists of IT infrastructure for providing web services. The Fog nodes are main components of computing. The Fog node is any device which offers virtual network connectivity, computing power, and storage capacity. These nodes can be used for machine learning, analysis, system optimization, and failure prediction [11, 12]. Figure 1 shows Fog nodes and its structure, where Fog computing deals with cloud data and IOT appliances [13].

Fog computing acts as a bridge between static and dynamic computing environments for a variety of industrial applications. The association of the Fog with various IOT applications such as smart city, smart healthcare, smart energy, smart manufacturing, and IOT devices is depicted in Fig. 2.

1.2 Industry 4.0

Industrial revolution has many phases among which industry 4.0 is the latest evolution which has brought development in various fields with emergence of smart devices. Industrial transformation, automation and exchange of data in manufacturing processes are collaboratively termed as Industry 4.0. The objective of industry 4.0 is to fulfil the requirements of customers as it has a huge impact on various areas like research and development, manufacturing, order management, and recycling of products. Figure 3 shows industry 4.0 as model that describes how industrial production has adapted latest developments and changes over time.

IIOT applications require storage capacity, high reliability, low latency, and real-time processing. This requirement is satisfied by using Fog computing [14, 15].

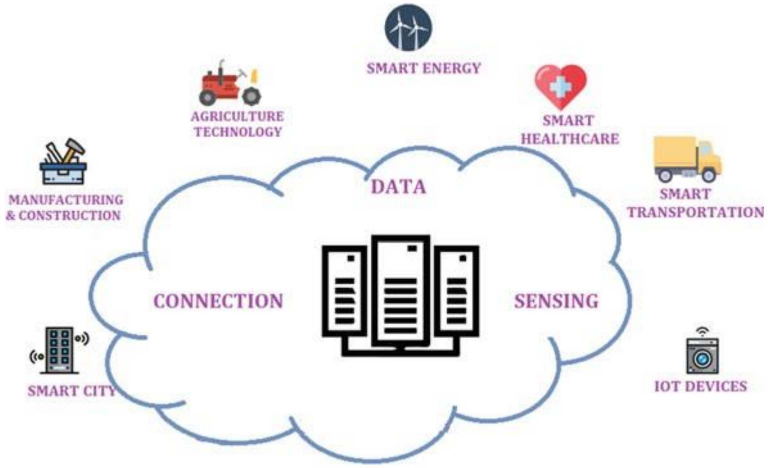


Fig. 2 The fog and IOT applications/devices

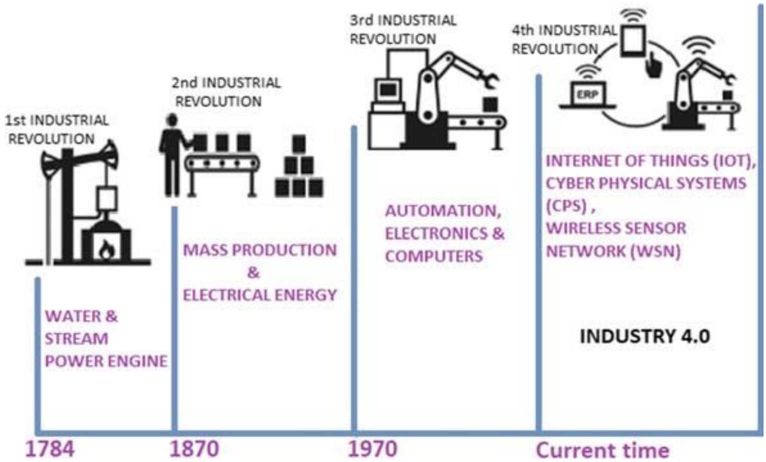


Fig. 3 Phases of industrial revolution

Industry 4.0 has contributed to all the domains comprising academics, economics, industry, and research areas [16]. Industry 4.0 has shown better results than its previous versions in terms of product development and manufacturing process with the incorporation of Fog computing [17]. Its environment consists of internet of services, internet data, internet of people, and internet of things. The collaboration of Industry 4.0 with smart infrastructure such as smart homes, smart logistics, social sites, and business applications has together developed CPS. This revolution has brought an opportunity for merging of real as well virtual worlds together on the basis of CPS [18]. Industry 4.0 has lead to development of big data, IOT, 5G, communication technologies, and Cloud/Fog/Edge/Mobile computing technologies [19–21].

2 Fog Architecture for Industrial Processes

Fog Computing is comparatively a new technology augmented in industrial applications. Many different definitions of the Fog Computing are given by researchers and technologists [2]. Even by few researchers, Fog Computing is also named differently. As per their own prospect, Cisco solutions are the leaders to define Fog Computing. They introduced Fog Computing as an expansion of Cloud Computing implemented at the boundary of the communication network. In [22, 23], Fog computing is also termed as Fog Networking. Its fundamental objective is to facilitate billions of associated smart devices to run a variety of applications at the edge of the network. In [24], Fog Computing is defined from a different prospect as an intermediate network amidst Cloud and smart devices. It is demonstrated that Fog Computing provides more optimized services as compare to Cloud Computing [25].

Recently introduced, the Fog Computing contributes in numerous IoT applications in a number of territories. Here, we present the Fog Computing architecture utilized in the different industrial IoT applications. Fog Computing provides a platform for the integration of a number of diverse objects which are termed as “Fog Nodes” and are unified with communication network architecture. This network is also equipped with routers to transmit the data packets and to be tailored with this network. The Fog Computing is also supported by programming interface to administer the needed data processing [26, 27].

A number of objectives are attained with different applications of Fog Computing in industry. In [24, 28], data transfer between IoT devices and Cloud are optimized by implanting the Fog servers amidst the terminal layer and the Cloud. Fog servers facilitate speedy transfer of the data gathered by different IoT devices which is the fundamental requirement in many real-time industrial applications. In the application discussed in [29, 30], authors stored data at data centers. The Fog servers facilitate data processing and data transfer to the Cloud. In [31], authors demonstrated the Fog architecture as an amalgamation of IoT devices, Fog Nodes, and Cloud computing layer. In the proposed architecture, a huge volume of data from industrial IoT devices is gathered by the Fog Nodes and comprehensive analysis is done by the Cloud Computing layer. Deriving from the work done by different authors, Fog Computing is based on multi-tier architecture to provide required resilience to the different applications [32]. But, it has been observed that the fundamental three layers are required in all the applications, as depicted in Fig. 4.

The bottom layer, comprising of various intelligent things like sensors, actuators, data processing devices, controlling systems, manufacturing systems, etc. interact with the middle layer to assure faster data processing. The middle layer is also addressed as the Fog Computing layer interfaces Fog Nodes and different routing devices. The knowledge extracted from the data at the middle layer is passed to the

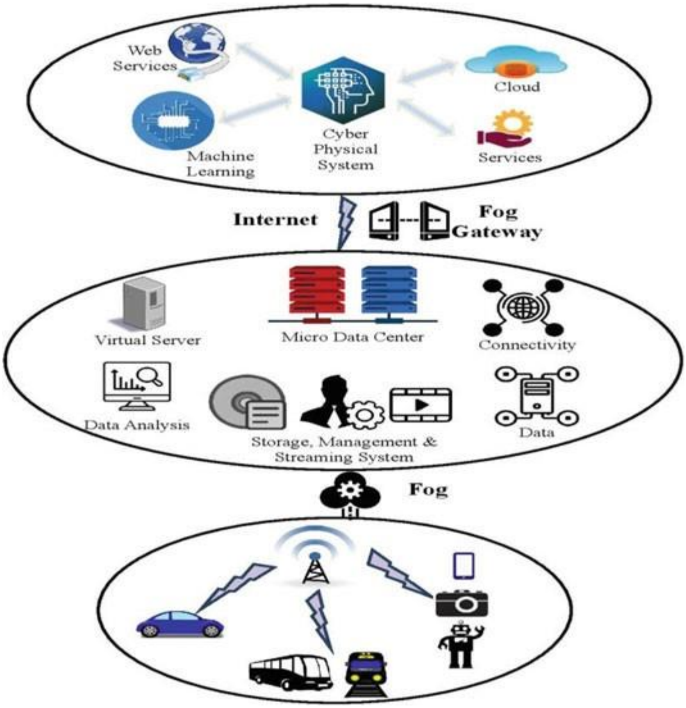


Fig. 4 Architecture-fog computing

topmost layer, i.e., Cloud Computing server for deep analysis and decision-making purposes [33, 34].

From the Fog architecture, it has been realized that Fog Computing is the most essential layer to support different services and applications for different users. In [35, 36], authors discussed that Fog Computing also helps in data optimization to be stored on the Cloud by eradicating irrelevant data. Fog computing also helps to defeat Cloud drawbacks. Fog plays a very vital role in temporary storage, preprocessing, privacy, and security of data administered in the industrial environment.

The Fog Computing is characterized by many important parameters. It facilitates the tasks of data storage, data processing, and computation [37]. Fog computing is also facilitated by very important computational features, i.e., scalability and resilience. It supports distributed data processing. Authors in [22, 29, 38], impressed upon the mobility and distributed architecture of Fog. The distributed architecture of Fog allows speedy data transmission over the communication networks to aid real-time industrial computations. The feature of mobility helps to avoid the transmission of the same data repeatedly to multiple Fog Nodes by using a special discovery algorithm to locate Fog Nodes [39]. Fog framework is also characterized for real time analysis of the huge volume of data near to its source which contributes to manage delays and latency and make it suitable for different applications introduced for industrial processes [29, 40]. The Fog computing supports versatility of nodes with different features and environments. This diversity needs to facilitate optimum interfacing of heterogeneous devices [33, 38]. Fog platform also assures privacy and security of data in different industrial applications [41].

3 Fog Equipped Industrial IoT

The complete advantage of Industrial Internet of Things (IIoT) can be taken only by using some suitable middleware like Fog. Fog supports resource utilization, data transfer among underlying nodes and facilitates for needed computational requirements [1]. All IIoT applications usually comprise of a variety of things like devices, sensors, actuators, etc. A Fog-supported IIoT architecture is depicted in Fig. 5.

Broadly, Fog can support a number of different tasks/activities in an industry according to the requirement. Fog can act as a micro or nano datacenter, cloudlet and even an edge device in a variety of situations. In crucial time, Fog can even take up the complicated tasks performed by the sensors. Similarly, Fog can also administer the energy usage and rate of data propagation of the sensors. Fog can oversee solar and thermal sources of energy too.

A number of IoT solution providers are available in the market. They have their own dominion to provide quality interoperability among the heterogeneous devices

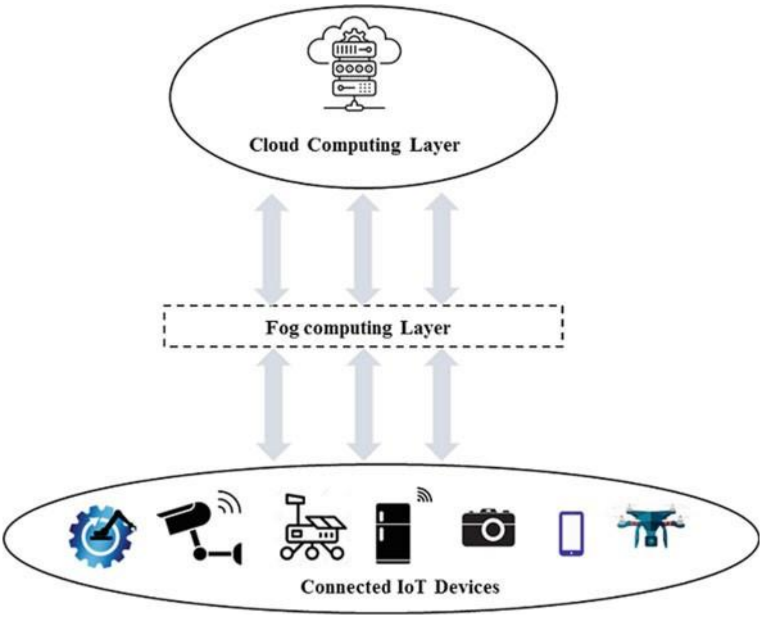


Fig. 5 Fog facilitated IIoT architecture

of the systems. Fog is equipped with a number of functions, protocols, and interfaces to address interoperability challenges [42]. Fog also plays a very imperative role to optimize different business operations. In IIoT, mechanization of complicated tasks is done with the help of robots. Consequently, middleware needs to be malleable and potent to administer robots [43].

Fog computing can perform efficiently in all the three domains of industry, i.e., extraction, manufacturing, and service. All the industrial verticals have varying competence to adopt IoT.

Here, we present few domains where IoT can be incorporated to accomplish thought objectives from IoT-supported future industries.

3.1 Transportation

Transportation is one of the most important services and acts as a backbone of a nation. All the trains, busses, heavy-duty vehicles, and private vehicles are the diverse elements of the transportation system. Intelligent transportation system (ITS) is an application of IoT which can facilitate Internet of Vehicles (IoV) using Fog. Using Fog, different services related to entertainment, location of the vehicle, road conditions, road traffic management, etc. can be supported in IoT equipped ITS.

3.2 Smart Grids

The smart grids have become very popular in the last decade. They comprise of smart meters, renewable energy resources, energy-optimized equipments, and intelligent appliances [44, 45]. With the technological advancements and industrial automation, the availability of smart appliances is increasing speedily. Consequently, a huge volume of data is generated on regular bases in the smart grids which need to be processed, analyzed, and stored very efficiently. Fog computing can play a very crucial role to administer various processes related to gathered data.

3.3 Mining

Data analytics is highly required in the Mining industry. As per IBM1, every person needs a huge volume of resources tentatively 3.11 million pounds of minerals and fuel for survival during the lifetime. Appliance of sensors and similar technologies in mines contribute to enhance the productivity keeping the cost on the lower side. The usage of automated and IIoT equipped instruments and IoVs also help to keep the process of mining cost effective. Few mining activities, e.g., chemical reactions are also dangerous in nature. This very problem can be overcome with the use of the sensors. Accuracy of the various processes involved in the domain can be increased using Fog computing. Maintenance of the various devices and energy efficiency are two most important parameters in the mining industry. Optimal usage of the machinery can also be attained in the mining industry using IIoT.

3.4 Agriculture

Sensor networks are already playing a very vital role in the domain of agriculture. The concept of smart farming is realized using sensors and smart equipments for various activities like seeding, watering, spraying, harvesting, etc. Fog equipped devices like drones can perform very efficiently in the tasks like monitoring the crops, spraying, and watering.

3.5 Food Industry

Packaging and storage of the food, quality analysis, frozen food, and inspection of organic waste all belong to food industry. Data analytics in the IIoT facilitated food industry gives birth to the modern food industry with high degree of quality control and waste management.

3.6 Waste Management

One of the fundamental goals of the waste management industry is to automate the process [46]. The process of waste management controls the spread of many diseases if done timely. Waste management task is also related to the recycling industry. In this process, there is always a requirement of resource allocation also to manage different types of waste [47]. Other details related to the area are available in [48].

3.7 Parking

Along with smart parking discussed by the authors in [49], it can be improved from a number of aspects. The various angles from which Fog and IIoT facilitated parking can be thought of are local festivals, weather conditions, hours of the day, etc.

4 Fog Computing in Industry 4.0

Industry 4.0 has made it possible for converging information technology to an environment of CPS along with advancement in technologies. These technologies have transformed traditional manufacturing systems to digital manufacturing systems. This has been achieved by collaborating computer systems along with industrial machine learning systems. Development in technology has revolutionized industrial manufacturing and made it digitized system. Figure 6 shows the different advanced technologies associated with Industry 4.0.

Various Fog facilitated application areas of Industry 4.0 are discussed below.

4.1 Industrial Internet of Things

IOT is considered to be the heart of Industry 4.0. IOT refers to the interconnection of physical devices over a network. IOT facilities exchange of data via internet. Smart devices are used actively nowadays in various fields like homes, healthcare centres, and education system. Smart phone is an important component for managing various online services such as health care applications, traffic analysis, etc. IIOT is a subset to IOT. Integration of sensors, electronics, and software with industrial machines has shown wonderful performance by collecting real-time data for analysis and predictions with the introduction of the Fog computing.

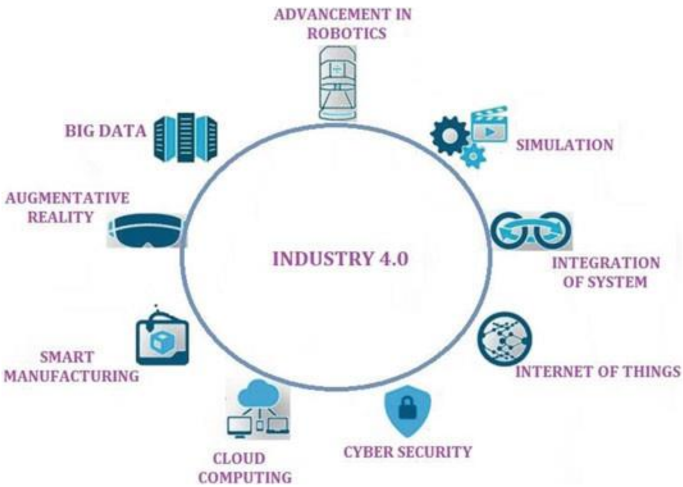


Fig. 6 Advanced technologies in Industry 4.0 applications

4.2 Big Data

Big Data refers to the large data which is extracted from various sources such as social media applications, sensors, cameras, computers, etc. In industrial manufacturing this data is considered to be captured using sensors and ERP systems. Conversion of this big data into useful information requires data analytics processes which is fastened using Fog computing. Machine learning and data visualization are two basic techniques used for management of this data by applying powerful computational algorithms.

4.3 Cloud Computing

Cloud computing is a platform for the users to process large amount of data on the remote servers. Cloud computing provides large storage space which helps in various business processes by usage of data analytics. Integration of Fog computing with cloud services decreases the communication delays which help in various manufacturing as well as business operations.

4.4 Advancement in Robotics

Industry 4.0 has brought revolution in the field of technology. Advancement in technology has made it easy to perform delicate and difficult tasks. Software and sensors can analyze and react to information received from the humans using Fog speedily and accurately.

4.5 Smart Manufacturing

In addition to robotics, 3D printing is also considered as a key part of Industry 4.0. It has provided the service of digital manufacturing technology. Incorporation of Fog with this approach has reduced transportation charges as well as simplified the process of storing digital files instead of physical piles.

4.6 Flexibility in Machines

Manufacturing has become an autonomous process with the usage of Industry 4.0 methodologies. This has introduced modern approaches such as “plug and produce”.

This will encourage development of intelligent infrastructure. Such type of manufacturing will encourage dynamic manufacturing lines along with speedy maintenance due to the incorporation of Fog computing.

4.7 Smart City Applications

Advancement of technology and growth in population has resulted to originate more data. This has lead to increasing demands for Fog services. For conversion of a city to “Smart City”, there is a need for transformation in various domains such as healthcare, smart homes, and agriculture. All of these domains have different types of requirements. Though development can be seen in many areas which comprise of smart living, smart people, and smart mobility, but yet this transformation requires more services.

4.8 Smart Factory Applications

Smart industry consists of distributed automated robotics. Fog computing, machine learning, and artificial intelligence are the most important techniques which can be used for producing smart factories. For evolution of the industry, industrial processes require cyber physical systems as well as IOT technologies.

4.9 Predictable Maintenance

Wireless connectivity and sensors together can monitor data easily by using machine learning approach to identify latest trends. Instead of using traditional approach of maintenance, engineers can use modern approach by analyzing patterns of equipments. This analysis can be done by observing frequency patterns of equipments in terms of breakage, its usage capacity, and errors by detecting problems in them before usage.

4.10 Augmentative Reality

Industry 4.0 has done its beginning to explore the benefits of augmentative reality. The biggest advantage of augmentative reality is of superimposing virtual image on physical data objects. To achieve this, smart phones, smart glasses, and smart tablets can be used. This has improved the method of decision-making using Fog.

5 Research Challenges

During literature survey, it has been observed that the industrial implementation of Fog faces a number and variety of challenges depicted in Fig. 7. Some of these challenges are elemental and others are dependent on them [2]. Heterogeneity, interoperability, programmability, and security are the fundamental challenges faced during Fog implementation whereas Quality of Service (QoS), cost, and power consumption are dependent on fundamental issues. In this section, we will discuss all these issues one by one.

5.1 *Heterogeneity*

In industry, always a number of heterogeneous systems, equipments, and smart devices are interconnected which generate real time heterogeneous data continuously. With the deployment of the Fog layer, smart devices become facilitated with extra storage and computational power [9]. But if compared with Cloud, the competence of Fog to manage diverse data in real time is much limited. It also lags behind to provide the equal QoS in managing real time data.

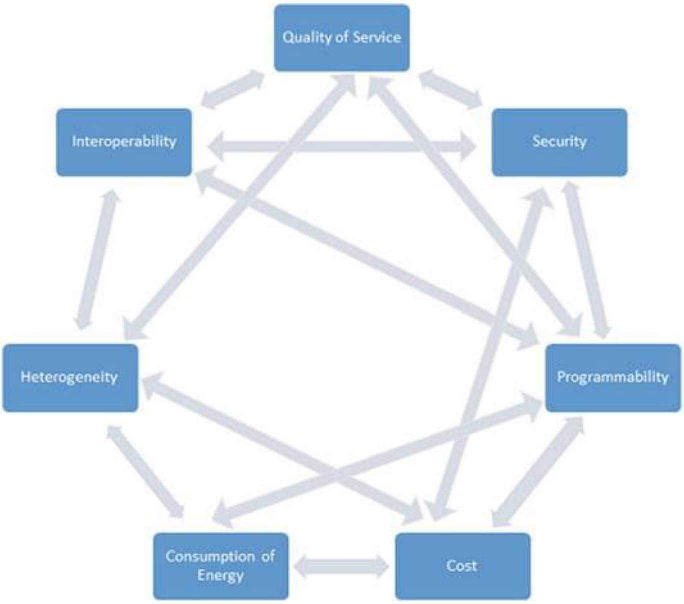


Fig. 7 Challenges: industrial implementation of fog

5.2 Security

Data storage and transmission are the fundamental tasks in any Fog based application. Security of the data either in storage or transmission is the most elementary issue faced in Cyber physical Systems (CPS) in the industry. As Fog is based on a decentralized architecture; so, to ensure data security is one of the most crucial aspects particularly deploying data security protocols on different Fog nodes.

5.3 Programmability

A programmable platform is the most fundamental need of the continuously evolving industrial environment. In [9], authors have proposed the need for flexibility in the Fog environment for the inclusion and exclusion of the different modules time to time as per the requirement to make it more suitable for industrial usage. In [35], some industry-friendly frameworks for the Fog architecture are proposed by the authors.

5.4 Interoperability

In industrial applications of IoT, number heterogeneous devices are assimilated. Consequently, interoperability is one of the major challenges faced in the automated industry. This issue can be overcome by implementing the Fog in industrial environment [41]. Deploying Fog computing layer permits the integration of more heterogeneous systems to be interfaced for efficient working [9].

5.5 Energy Consumption

To optimize power consumption is also one of the important goals to be achieved in 4^{th} industrial revolution using Fog computing which will reduce energy cost. The reason behind this is the more power usage in Fog environment as compared to the centralized Cloud platform. So, fog architecture need to be optimized for lower energy usage.

5.6 Quality of Service (QoS)

Various quality parameters in an industrial environment are performance, data transfer in real time, connectivity, etc. [41]. The Fog comprises of heterogeneous nodes with varying computational power providing different levels of QoS for different industrial applications. To normalize, the QoS delivered using Fog computing is also one of the crucial challenges faced in Industry 4.0.

5.7 Cost

Success of any new technology in the industry is characterized by its cost also. Cost of Fog nodes is also a big hindrance to its acceptance in the industry which needs to be addressed with appropriate business model.

All the above issues need to be given serious thought for the successful deployment of Fog in Industry 4.0. Researchers need to take up these issues to pursue their work to administer these challenges and aid the industrial applications to reap maximum benefit from the integration of Fog in different industrial processes and applications.

6 Conclusions

Latest developments in the fields of IoT, network and communication technologies, and robotics have set a roadmap for the automation of the various industrial processes. Industry 4.0 and IIoT together have a huge scope of developments but it lead to a variety of challenges too. But with sophisticated planning and order these challenges can be easily administered and can be seen as new opportunities to be worked upon.

In this chapter, in the beginning the concepts of the Fog and Industry 4.0 are introduced to the readers. It is presented that how data is sensed by the Fog nodes and processed afterwards. Role of the integration of IoT and Fog to facilitate the functioning of various devices and applications in the areas of agriculture, healthcare, smart cities, transportation, manufacturing and construction, etc. is also discussed. It has been discussed that how Fog computing can contribute to make the industrial processes more efficient. We also present the different challenges like interoperability, programmability, energy consumption, QoS, etc. faced during the Fog deployment in Industry 4.0 which provides a direction to the researchers to carry on their work in the domain of industrial implementation of the Fog computing.

References

1. Aazam M, Zeadally S, Harras KA (2018) Deploying fog computing in industrial internet of things and industry 4.0. IEEE Trans Industr Inf 14:4674–4682
2. Bouzarkouna I, Sahnoun M, Sghaier N, Baudry D, Gout C (2018) Challenges facing the industrial implementation of fog computing. In: 2018 IEEE 6th international conference on future internet of things and cloud (FiCloud), pp 341–348
3. Goyal LM, Mittal M, Sethi JK (2016) Fuzzy model generation using Subtractive and Fuzzy C-Means clustering. CSI Trans ICT 4:129–133
4. Kagermann H, Wahlster W, Helbig J (2013) Recommendations for implementing the strategic initiative Industrie 4.0: final report of the Industrie 4.0 Working Group. Forschungsunion: Berlin, Germany.
5. Mittal M, Balas VE, Goyal LM, Kumar R (eds) (2019) Big data processing using spark in cloud. Springer, Berlin
6. Mittal M, Sharma RK, Singh VP (2014) Validation of k-means and threshold based clustering method. Int J Adv Technol 5(2):153–160
7. Peralta G, Iglesias-Urkia M, Barcelo M, Gomez R, Moran A, Bilbao J (2017) Fog computing based efficient IoT scheme for the Industry 4.0. In: 2017 IEEE international workshop of electronics, control, measurement, signals and their application to mechatronics (ECMSM), pp 1–6
8. Saleem MA, Zhou S, Sharif A, Saba T, Zia MA, Javed A et al (2019) Expansion of cluster head stability using fuzzy in cognitive radio CR-VANET. IEEE Access 7:173185–173195
9. Vaquero LM, Rodero-Merino L (2014) Finding your way in the fog: towards a comprehensive definition of fog computing. ACM SIGCOMM Comput Commun Rev 44:27–32
10. Barcelo M, Correa A, Llorca J, Tulino AM, Vicario JL, Morell A (2016) IoT-cloud service optimization in next generation smart environments. IEEE J Sel Areas Commun 34:4077–4090
11. Christian M (2018) Fog computing. Bus Inf Syst Eng 60:351–355
12. Mittal M, Battineni G, Goyal LM, Chhetri B, Oberoi SV, Chintalapudi N et al (2020) Cloud-based framework to mitigate the impact of COVID-19 on seafarers' mental health. Int Marit Health 71:213–214
13. Harish G, Nagaraju S, Harish B, Shaik M (2019) A review on fog computing and its applications. Int J Innov Technol Explor Eng (IJITEE), pp 2278–3075
14. Basir R, Qaisar S, Ali M, Aldwairi M, Ashraf MI, Mahmood A et al (2019) Fog computing enabling industrial internet of things: state-of-the-art and research challenges. Sensors 19:4807
15. Singh R, Gehlot A, Khilrani JK, Mittal M (2020) Internet of things–triggered and power-efficient smart pedometer algorithm for intelligent wearable devices. In: Wearable and implantable medical devices. Elsevier, pp 1–23
16. Lu Y (2017) Industry 4.0: a survey on technologies, applications and open research issues. J Ind Inf Integr 6:1–10
17. Ladiges J, Fay A, Holm T, Hempen U, Urbas L, Obst M et al (2017) Integration of modular process units into process control systems. IEEE Trans Ind Appl 54:1870–1880
18. Vogel-Heuser B, Diedrich C, Pantförder D, Göhner P (2014) Coupling heterogeneous production systems by a multi-agent based cyber-physical production system. In: 2014 12th IEEE international conference on industrial informatics (INDIN), pp 713–719
19. Roblek V, Meško M, Krapež A (2016) A complex view of industry 4.0. SAGE Open 6:2158244016653987
20. Wan J, Tang S, Shu Z, Li D, Wang S, Imran M et al (2016) Software-defined industrial internet of things in the context of industry 4.0. IEEE Sens J 16:7373–7380
21. Gruber FE (2013) Industry 4.0: a best practice project of the automotive industry. In: IFIP international conference on digital product and process development systems, pp 36–40
22. Stojmenovic I, Wen S (2014) The fog computing paradigm: scenarios and security issues. In: 2014 Federated conference on computer science and information systems, pp 1–8
23. Arora V, Leekha RS, Lee K, Kataria A (2020) Facilitating user authorization from imbalanced data logs of credit cards using artificial intelligence. Mob Inform Syst 2020

24. Aazam M, Huh E-N (2014) Fog computing and smart gateway based communication for cloud of things. In: 2014 International conference on future internet of things and cloud, pp 464–470
25. Bonomi F, Milito R, Zhu J, Addepalli S (2012) Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on Mobile cloud computing, pp 13–16
26. Hong K, Lillethun D, Ramachandran U, Ottenwälder B, Koldehofe B (2013) Mobile fog: a programming model for large-scale applications on the internet of things. In: Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing, pp 15–20
27. Kataria A, Ghosh S, Karar V (2018) Data prediction of optical head tracking using self healing neural model for head mounted display
28. Gupta H, Vahid Dastjerdi A, Ghosh SK, Buyya R (2017) iFogSim: a toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. Software: Pract Exp 47:1275–1296
29. Aazam M, Huh E-N (2015) Dynamic resource provisioning through fog micro datacenter. In: 2015 IEEE international conference on pervasive computing and communication workshops (PerCom workshops), pp 105–110
30. Rani S, Gupta O (2017) CLUS_GPU-BLASTP: accelerated protein sequence alignment using GPU-enabled cluster. J Supercomput 73:4580–4595
31. Wu D, Liu S, Zhang L, Terpenny J, Gao RX, Kurfess T et al (2017) A fog computing-based framework for process monitoring and prognosis in cyber-manufacturing. J Manuf Syst 43:25–34
32. Mouradian C, Naboulsi D, Yangui S, Glitho RH, Morrow MJ, Polakos PA (2017) A comprehensive survey on fog computing: State-of-the-art and research challenges. IEEE Commun Surveys Tutorials 20:416–464
33. Mukherjee M, Matam R, Shu L, Maglaras L, Ferrag MA, Choudhury N et al (2017) Security and privacy in fog computing: Challenges. IEEE Access 5:19293–19304
34. Bonomi F, Milito R, Natarajan P, Zhu J (2014) Fog computing: a platform for internet of things and analytics. In: Big data and internet of things: a roadmap for smart environments. Springer, Berlin, pp 169–186
35. Dastjerdi AV, Gupta H, Calheiros RN, Ghosh SK, Buyya R (2016) Fog computing: principles, architectures, and applications. In: Internet of things. Elsevier, pp 61–75
36. Rani S, Gupta O (2016) Empirical analysis and performance evaluation of various GPU implementations of Protein BLAST. Int J Comput Appl 151
37. Mukherjee M, Shu L, Wang D (2018) Survey of fog computing: fundamental, network applications, and research challenges. IEEE Commun Surveys Tutorials 20:1826–1857
38. Osanaiye O, Chen S, Yan Z, Lu R, Choo K-KR, Dlodlo M (2017) From cloud to fog computing: a review and a conceptual live VM migration framework. IEEE Access 5:8284–8300
39. Soo S, Chang C, Loke SW, Srirama SN (2018) Proactive mobile fog computing using work stealing: data processing at the edge. In: Fog computing: breakthroughs in research and practice. IGI Global, pp 264–283
40. Gupta O, Rani S (2013) Accelerating molecular sequence analysis using distributed computing environment. Int J Sci Eng Res–IJSER
41. Chang C, Srirama SN, Buyya R (2017) Indie fog: An efficient fog-computing infrastructure for the internet of things. Computer 50:92–98
42. Cañas C, Pacheco E, Kemme B, Kienzle J, Jacobsen v (2015) Graphs: a graph publish/subscribe middleware. In: Proceedings of the 16th annual middleware conference, pp 1–12
43. Monajjemi V, Wawerla J, Vaughan R (2014) Drums: a middleware-aware distributed robot monitoring system. In: 2014 Canadian conference on computer and robot vision, pp 211–218
44. Depuru SSSR, Wang L, Devabhaktuni V, Gudi N (2011) Smart meters for power grid—challenges, issues, advantages and status. In: 2011 IEEE/PES power systems conference and exposition, pp 1–7
45. Farhangi H (2009) The path of the smart grid. IEEE Power Energ Mag 8:18–28
46. Shukla R (2016) Smart waste management. Asian J Pharm Educ Res 5:48–54

47. Glouche Y, Couderc P (2013) A smart waste management with self-describing objects. In: The second international conference on smart systems, devices and technologies (SMART'13)
48. Aazam M, St-Hilaire M, Lung C-H, Lambadaris I (2016) Cloud-based smart waste management for smart cities. In: 2016 IEEE 21st international workshop on computer aided modelling and design of communication links and networks (CAMAD), pp 188–193
49. Andreev S, Galinina O, Pyattaev A, Gerasimenko M, Tirronen T, Torsner J et al (2015) Understanding the IoT connectivity landscape: a contemporary M2M radio technology roadmap. IEEE Commun Mag 53:32–40

Fog Computing Based Architecture for Smart City Projects and Applications

Naishadh Mehta, Anand Ruparelia, and Jai Prakash Verma

Abstract Fog computing is an extension to cloud computing, offering benefits such as minimal latency, wide geographical distribution, and location awareness by providing flexible services at the edge of the network. The onset of fog computing has catered solutions to many applications, Smart City projects being one of them. Fog computing has the potential to deliver an impact in smart city projects, as the former application involves economic and social aspects along with the technical aspect. The increase in city urbanization demands smart solutions that tackle critical problems such as healthcare, mobility, infrastructure, parking space availability, waste management, and energy consumption. Industry 4.0 conceptualizes that, Internet of Things (IoT) along with fog computing would be used for the development of a network of devices. These devices function independently in real-time and provide the required infrastructure for a smart city. This research study presents a comprehensive literature survey on the deployment architectures of fog computing in smart city applications such as Smart Waste Management and Smart Parking. An emphasis is laid more on the integration of Industry 4.0's core concepts and fog computing while also taking into consideration the deployment aspects. With the proposed architectures and mentioned approaches, improvements would be seen in terms of resource utilization, processing overhead, and latency. In the latter part of the research survey, the potential merits of the proposed approaches and future work directions are discussed.

Keywords Fog computing · Smart city projects · IoT (internet of things) · Smart parking · Waste management · Industry 4.0

N. Mehta · A. Ruparelia · J. P. Verma (✉)
Institute of Technology, Nirma University, Ahmadabad, India
e-mail: jaiprakash.verma@nirmauni.ac.in

N. Mehta
e-mail: 19mced07@nirmauni.ac.in

A. Ruparelia
e-mail: 19mced12@nirmauni.ac.in

R. Tiwari et al. (eds.), *Energy Conservation Solutions for Fog-Edge Computing Paradigms*,
Lecture Notes on Data Engineering and Communications Technologies 74,
https://doi.org/10.1007/978-981-16-3448-2_10

1 Introduction

With the rise of population around the world, the rate of urbanization in modern cities is also increasing. According to a report by the United Nations, the urban population (worldwide) is expected to reach as much as 70% of the world's population by 2050 [1]. This poses the critical challenges for cities to provide proper civic amenities [2] such as water supply, healthcare, public infrastructures, etc. to all the citizens. The concept of smart cities was introduced long back with the aim of providing a sustainable environment to all the residents, thereby providing a decent quality of living. The concept has now reached a stage, where all the facilities needed for its implementation are developed. An important one amongst them is IoT (Internet of Things).

Figure 1 shows the global share of IoT projects in different regions across the world in 2018, where smart city ranks in the first position. Well, the first position was held by the Connected Industry in 2016 [4]. But, strong initiatives by different national governments across the world, towards Smart Cities, made it secure the first position in 2018. The major share of IoT projects in Europe is towards Smart City whereas, in the United States of America, the Connected Health projects are taking the lead. The Asia Pacific region (APAC) focuses more on IoT projects for Smart Agriculture. In the Smart City projects, various segments such as Smart Traffic that handles parking systems, traffic monitoring, smart bus lanes are considered for the development. The revolution in IoT acted as a catalyst in the growth and development of smart cities by bringing the term from mere thought to reality. As the advancements in technology are happening rapidly, there has been a remarkable increase in the commercial digital

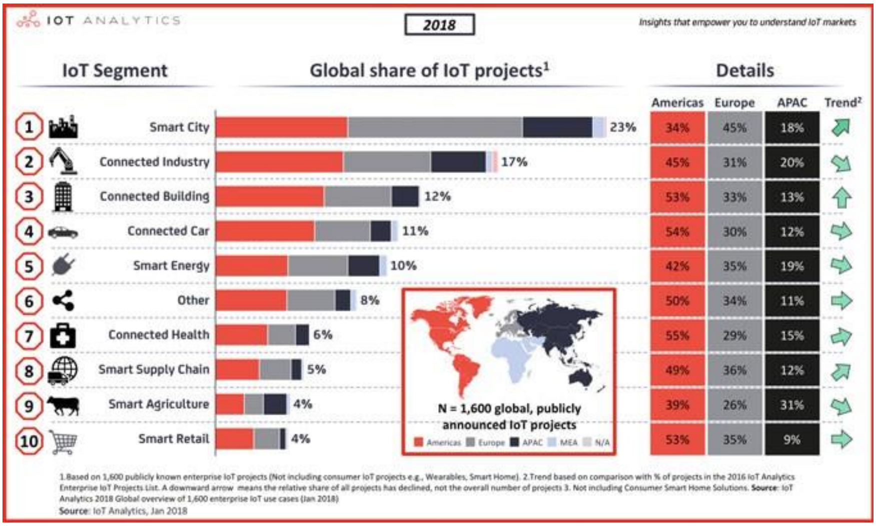


Fig. 1 Global share of IoT projects [3]

services consumed by the people in the form of digital devices such as smartphones, gadgets, wearable, and smart appliances. The interconnection of these digital devices goes hand-in-hand with the digital progression, where the Internet acts as a medium of communication between them which was a bit difficult in the previous era due to the fact that other services were not pervasive nor had ubiquitous computing facilities [5].

The ultimate goal of smart city projects is to develop a city that functions in an intelligent manner and also sustainable for a longer run to achieve this. An integration of different infrastructure modules is needed to monitor and control their operations. IoT plays the key role at this point as IoT based smart cities have many applications that can be technically classified into types of network, scalability, heterogeneity, flexibility, and coverage [6]. The general classification would be based on the end-user involvement such as personal, home, and enterprise as IoT is technology-oriented while smart cities are user-oriented [7]. It caters solutions to critical as well as daily use-cases such as safety management, education, healthcare, public transportation, environment, city space management, and many more. As per Fig. 2 shows the smart city market across the globe is on a rise.

Though, exponential growth in smart city development around the world is not expected. Instead, the steady growth of the smart city market is predicted which would exceed the global market by $2.5 trillion in 2025. With the market curve rising, an important aspect of smart cities comes into the picture. This deals with the roles and responsibilities of the key stakeholders such as government bodies and the

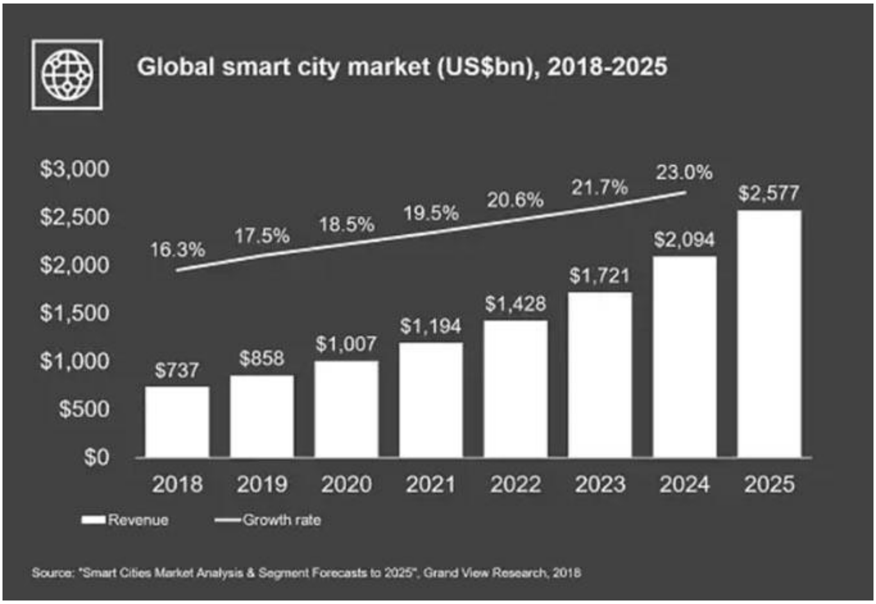


Fig. 2 Smart city market prediction [8]

private sectors that aid in fulfilling the aspirations of the project. Though the smooth relationship and small steps towards the goal would raise the market curve at a steady pace it would be a successful one.

The core network architecture of smart cities certainly includes a large number of physical devices which are connected to the Internet. As the intercommunication of these devices happens, a huge volume of data is generated [9], and that leads to “Big Data” which requires efficient storage and when the whole city is involved, there are millions of devices connected together and the generated data traffic is pretty much extensive than the conventional traffic which needs to be handled by a hardware environment that is capable enough to store, manage and process the data in real-time and for that reason, cloud computing comes into the picture [10]. Smart cities empowered with IoT technologies utilize cloud computing for data management as the interaction within the network of different devices is totally dependent on the data being generated and exchanged at incompatible rates and thus it makes a necessity for this paradigm to enable data-related services to run in apt environment and also to offer as consistent latency as possible.

1.1 Fog Computing and IoT

IoT made conventional communication more efficient and the cloud computing handles the issues of storage, processing, and efficiency of the devices involved in the IoT. Hence, It has been very helpful in building up many applications relating to healthcare, agriculture, environment protection, industrial automation, as part of the case for smart city projects. As previously mentioned, the concept of the latter has been more in research areas these days due to the central theme of IoT in it [7]. It has the potential to deliver effective solutions. Needless to say, the concept itself is a big umbrella under which many components do exist, all of them primarily holded by their “digital nature”. As a result, the term is now aging beautifully and becoming more smarter than before [11]. It blends from the connection of key industry and service areas such as governance, mobility, utilities, buildings, and environment [12]. Along with the benefits it possesses, there are several complexities also associated with the same. It is an ecosystem that is characterized by heterogeneity in terms of technology being realized and employed in the concept of Fog Computing. Few characteristics of the fog computing’s are as follows [13]:

1. ***Location information and low latency***: The main advantage of fog computing is it’s support of location awareness through which, we can deploy fog nodes in different locations. As, the distance between fog nodes and end devices decreases, the latency is decreased for processing the end devices data.
2. ***Geographical area handling***: The services and applications offered by centralized cloud computing paradigm were not distributed. So, in that respect, fog computing covers the geographical distribution of fog nodes in a distributed manner which is efficient for deployment purposes.

3. ***Scalability***: Scalability of fog computing is defined by the distributed computing and storage services offered in a large-scale network of devices. This makes dynamic addition and removal of devices as per the requirement of the application.
4. ***Mobility help***: The ability of fog applications to connect aptly with mobile devices enables various mobility methods such as LISP (Locator ID Separation Protocol) that requires a directory system which is distributed.
5. ***Interactions in real-time***: Integrating cloud computing without the extension of fog computing in any application, performs batch processing interactions while fog computing offers real-time interactions between the fog nodes.
6. ***Heterogeneous nature***: Different vendors and manufacturers design and develop fog nodes or end devices and hence there is always a need for customized deployment suiting to their platforms. Fog has the ability to handle the heterogeneity smoothly.
7. ***The interoperability***: With heterogeneity, there is a demand of inter operating and working with various domains spread across ample service providers. This interoperability is also one of the nature of fog components.
8. ***Support for web analytics and cloud interplay***: The natural position of fog is between the cloud and end devices. This helps in efficient processing and analytics of the data in proximity of the end devices.

1.2 Fog Computing Versus Cloud Computing

Cloud computing provides many benefits such as remote resource management and also eliminates the hassle of managing the data locally. There are many services and applications which are getting ample benefits from cloud computing facilities. With the consideration of increases the use of cloud computing in many applications of IoT and also due to systemic nature of cloud services, several issues may raised such as inconsistent latency, lack of mobility support, and location-awareness, these problems need to be addressed. Cloud computing also has few challenges that revolve around scalability, those also need to be considered [14]. Amongst these, the latency related issues are major challenges faced by few time-critical applications and services [15]. Therefore, a technology that solves these issues as well as aids in catering to better smart cities, needs to be adopted and Fog Computing is one such technology.

Fog computing, also known as edge computing, will overcome these difficulties by offering elastic computing at the edge of the network, where resources and services are accessible to end-users quickly with lesser delays, while cloud computing is all about having resources spread across the core network [16]. Yi et al. [16] defined fog computing as geographically distributed architecture with a resource pool consisting of one or more pervasive devices present at the edge of the network in isolated locations. Through which, they collaboratively provide elastic computing, storage, and connectivity to a massive scale of relevant devices. Well, Fog Computing canâe™t be considered as a complete alternative cloud computing, in one way, it is considered as a building block for the cloud services [17]. By making the communication, computing and processing capabilities very near to the end points, the optimization in usage of network resources is possible [18].

1.3 Contribution

In this research study, we've presented, the amalgamation of Industry 4.0, Smart Cities, and Fog Computing in a way that, they complement each other by offering a better solution to existing options.

1. A literature survey of various deployment architectures of fog computing is presented in the form of a comparison along with their merits and limitations.
2. Presented the current researches related to, Smart Waste Management and Smart Parking, the novel architectures for both Smart City applications are also proposed in the later part of the research study.

2 Related Work

Many researchers have worked in the area of fog computing in smart city projects [19]. Few of recent research work is presented in Table 1 with their findings and limitations. Zhang [20] proposed a solution of big data processing and network scalability problems for fog computing environment in a smart city project. The proposed scheme achieved, reduced task processing and delay. And presented that the proposed scheme is more suitable for the fog environment with more computing resources. Lin et al. [21] implemented a fog computing-based decision-making framework with the adaption of Convolutional Neural Network (CNN) deep learning model. It helps to improvise the calculation performance for monitoring the assembly lines of a manufacturing firm to identify the defective products. Naranjo et al. [22] presented a fog computing architecture network (FOCAN) for providing the services to IOT applications in efficient manner with low energy usages. This multi-tier network help to improve the efficiency of fog computing environment for a smart city project.Nasir et al. [23] proposed a novel Fog computing enabled distributed video summarization scheme for surveillance videos based on small, low-powered, cheap, single board computer known as Raspberry Pi. With a very little overhead the proposed framework is more efficient compare to traditional cloud computing environment. Srinivasa Desikan et al. [24] proposed algorithms for the two phase of fog computing network architecture. The algorithm applied for the first phase construction perform better compare NIGS and HTC y 45% and 42% respectively. The vacation-based Resource Allocation (VRA) algorithm proposed for second phase enhances the performance of resource utilization by 12 × times.

Table 1 Research work done in the area of Fog computing

References	Objective	Methodology	Findings	Limitations
Zhang [20]	Solve the issue of big data processing and network scalability for better management of data in a smart city project	Fog computing and Internet of Things service platform for smart city	Effectively reduce the task processing delay, task violation rate, and running time of the resource allocation	Lack of the optimization of computing resource allocation under the constraints of fog computing environment
Lin et al. [21]	To find possible defective products for large number of assembly Lines in a manufacturing firm	Fog Computing based Hybrid Deep Learning Framework (FC-HDLF)	To improve the performance in a fog computing environment using the Convolutional Neural Network (CNN) model	The issues of scalability and integration are not addressed
Naranjo et al. [22]	Form multi-tier network structure for applications based on joint compute, route, and communication through smart city environment	Fog-supported smart city network architecture	The devices can provide the services with low energy usage and in an efficient manner	The impact of the 5G technology is not considered with the presented Fog Computing Architecture Network (FOCAN)
Nasir et al. [23]	A novel resource-efficient framework for distributed video summarization over a multi-region fog computing paradigm	A cloud computing environment, where node is represented by Raspberry Pi systems	With very little overhead, proposed cloud computing framework is more cost-effective solution with good scalability	The proposed framework is not testing with different robust Fog Computing architectures
Srinivasa Desikan et al. [24]	A topology control technology for managing and constructing the large scale IOT network for smart city project	An efficient algorithm for construction phase and maintenance phase to optimize the resource utilization	The proposed algorithm performs 45% and 42% better compared to NIGS and HTC respectively. The Vacation based Resource Allocation (VRA) enhances resource utilization by 12 times	The self-organization capability of network can be achieved by applying the reinforcement learning

3 Smart City Projects in India

Over the last few years "Smart City" has been a talking point in the sectors of government authority, Industrial giants, Investors, Academia, research laboratories, and general mass of end users in India. Smart cities are new urban ideas that are key to the quality of life for individuals. In order to attain wise and sustainable activities, it is the philosophical perspective of grouping separate innovations [25]. After the industrial revolution, major cities have been the engines of economic development. Although effective in stimulating growth, economic development has not always been "smart" compromising human well-being for competitive advantage. Leading smart cities such as Stockholm, Barcelona, New York, Vienna, and Toronto have used technical advances to integrate efficiency into structures, facilities, social spaces, improving the livability, workability, and resilience of these areas [26]. Driven by these trends in smart cities, India plans to create 100 smart cities in different parts of the nation. India's Smart Cities Mission (SCM) is a strategic project declared in 2015 by the Ministry of Urban Development (MoUD) to create an infrastructure for 100 smart cities in upcoming years [26].The GOI follows an eligibility and selection criteria for smart cities. This process depends upon a competition [27] which is held across the nation. The competition flow for smart cities challenge is depicted in Fig. 3.

Success of IoT has been raised because of implementation of Smart city projects. The Government of India (GOI) has already released its draft policy through the Ministry of Electronics and Information Technology (MeitY) where it has been reported that IoT will open a new market model with massive revenues, and GOI will plan to spend around USD 15 billion in the IoT industry by 2020 when the number of IoT devices in India is around 2.7 billion [28].

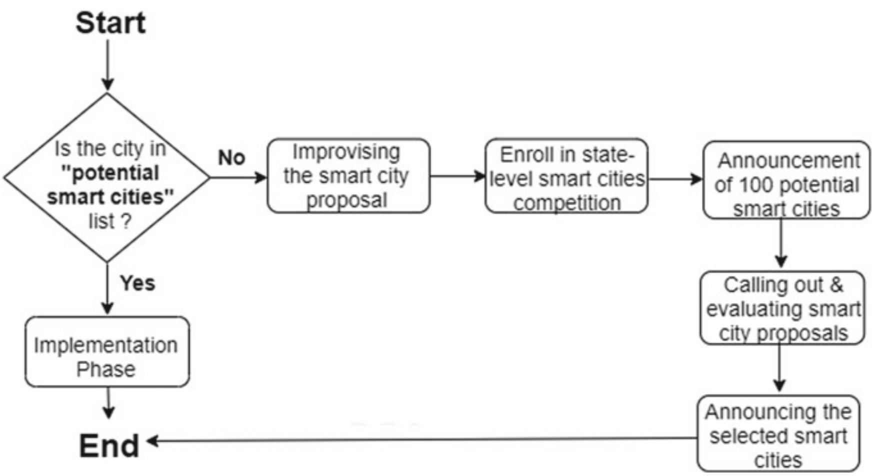


Fig. 3 Smart city selection process

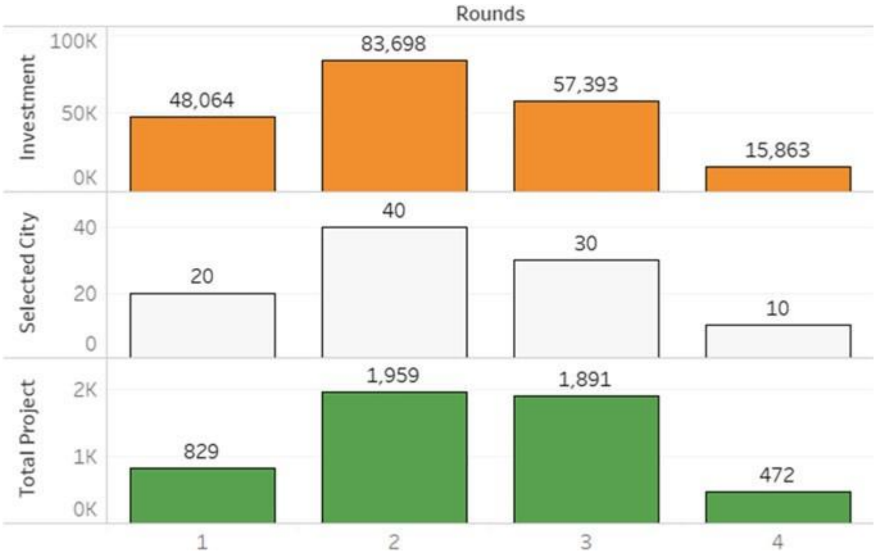


Fig. 4 Development of smart cities (round wise)

As shown in Fig. 4, the development of smart cities in India is done round-wise, and investments are done accordingly in different selected cities [27]. Sixty cities were currently chosen in the SCM (from the list of 100 planned smart cities) for initial investment in the introduction of SCP. More than half of the various provinces are situated in the states of Uttar Pradesh (13), Tamil Nadu (12), and Maharashtra (10). A special purpose vehicle has been set up in these 60 cities to evaluate the success of the mission at the city level [26]. Most of the green field projects throughout the name of continuous and resilient cities also have been constructed and established in India namely Lavasa, Gift City (Ahmedabad), Kochi Smart City, Nano City, and others like Dholera have been named Smart Cities [29]. However, in terms of resources and size, the speed of industrialisation and growth in the country seems far from sufficient, so there's an immediate need for growth in this aspect.

3.1 Industry 4.0 and Smart City Projects

The business is the piece of an economy that produces material products which are exceptionally motorized and automatized. From the advent of industrialization, mechanical strides have led to shifts in perception that are now "modern upsets": in the area of industrialization and urbanization (the presumed first modern transformation), of the expanded use of electrical resources (the presumed second mechanical insurgency) and of wide-ranging digitalization (the purported third modern unrest)

[30]. Centered on high-level digitization inside manufacturing plants, the combination of Internet advances and future-arranged developments in the area of “keen” objects (machines and items) appears to bring about another paradigm shift in the outlook of modern creation. The vision of future development includes isolated and efficient mechanisms for assembling and describes scenarios in which objects manage their own assembling period. This can be known by packaging individual products in a cluster size of one while retaining large-scale production financial states. Encouraged by this potential expectation, the word “Industry 4.0” was used ex-ante for the organized “fourth technological revolution".

Industry 4.0 is introducing a new concept for the convergence of digital and physical environments by cyber-physical networks [31]. Industry 4.0 primarily focuses on the faster speeds of content deliveries [32]. By the fourth Industrial revolution it helps to do such change in working environment by proposing new facilities, ideas, and very well-organized business models. Industry 4.0 models have also proposed significant benefits to Industry by enabling smooth manufacturing process and also create an environment to analyze the Big data in real time which are generating from manufacturing process to take critical and efficient business decisions. Upgradation in this industry, granted the development of integrated and interconnected systems. These systems can increase production for the firms with better and efficient use of resources with the help of smart grids for energy savings [33]. The entire definition of Industry 4.0 is based on six design principles. These standards help organizations in the detection and execution of Industry 4.0 strategies [34].

1. ***Integration***: Potential to connect individuals with Industry 4.0 to interact and engage with each other by using the Internet as assistance.
2. ***Federalization***: Increase in demand for service creates difficulties to manage the systems but by this they make their own decisions.
3. ***Virtualization***: By using this concept we can monitor the sensor data with computational models.
4. ***Real-time***: Analyses and generates meaningful insights from Big Data in real time and also reacts quickly on the errors or faults.
5. ***Orientation of service***: The services of human individuals or service providing firms or CPS (Cyber-Physical Systems) can be made available to other end users by serving the Internet as a Service.
6. ***Interoperability***: Adaptable transition of Industry 4.0 Production Plants to changing requirements by substituting or expanding individual modules just like changing requirements by substituting or increasing individual modules.

These two concepts focus in particular on the Internet of Things, the Internet of Energy, and the Internet of People. Company has consistently been a central part of the overall concept of metropolitan areas and cannot be viewed separately. Connecting singular pieces can be natural with the IoS (Internet of Services). A more essential aspect is FOG Computing, which transfers out critical data correctly, and the best part of the job is to try to do that dynamically, either in skillful devices or in the parent communication protocols The smart city relies on six main zones: Smart Citizens, Smart Economy, Smart Living, Smart Governance, Smart Connectivity and Smart Climate [34].

Industry 4.0 developments are building a radically modern "smart city" system. These emerging developments enable the challenges of resource usages and energy enhanced operational performance to coordinate urban development and population shifts in megacity. On the other hand, the construction of a smart city encourages new industrial revolution, provides new environments for livelihood, working, literacy, accrues personal and social capital, and draws monetary success for business growth. Smart City development with the use of Industry 4.0 can be used in fields of Transportation, energy systems, infrastructure development, business expansions, and developing all smart facilities to serve the community [35]. Many Smart City near to 4.0 frameworks is expected to function on their own inside a distributed existence with the capacity to interconnect and work together within a structured data gathering. Smart City near to 4.0 will go ahead in a joint loop with the whole Community, where this dynamic urban transformation is identified.

4 Role of Fog Computing in Industry 4.0 for Smart City Projects

The conventional approach of network architecture is based on cloud computing, collecting and sensing the data through the IoT devices/sensors and forwarding it to the cloud servers for processing and analysis. In the later stage, the results from the cloud servers are sent back to the users of the IoT devices [36]. The infrastructure is well-suited to the IoT applications where data processing capabilities are required at the forefront. Though, with increase in the number of IoT devices across the globe, this infrastructure suffers from issues such as network congestion and high server loads, which leads to latency issues. Hence, fog computing is employed, which essentially translates to hosting and operating on network ends [37]. Despite the fact, there are points to be considered while deploying the same in a real-time environment.

As shown in Fig. 5 deployment applications in a fog computing environment leverages â€œfog nodesâ€œ in the data centers, network edges, and also at the end devices. The management of these nodes is completely dependent on the centralized server, which handles the data requests from the users and decides on serving those on fog nodes. These fog nodes help to gather the important data and pre-process it locally on them, at a lower level than cloud. This platform ultimately extends the cloud platform to the end devices which are near to the end users, helping to minimized the latency [17]. The deployment of fog computing architecture is complex and differs from application to application. Hence, the initial step for identifying the best approach for proposing any architecture for smart city project is to study various fog deployment models proposed by different researchers shown in Table 2.

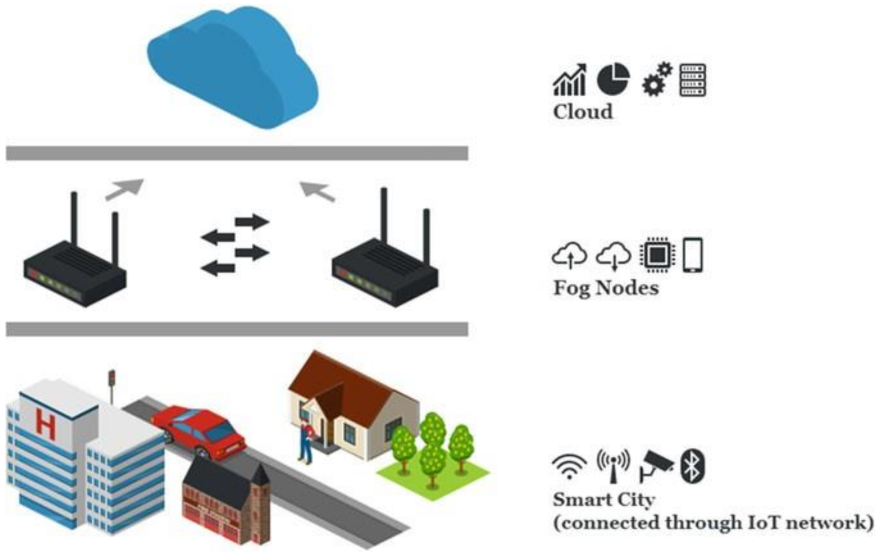


Fig. 5 Smart cities architecture using fog computing

5 Fog Computing Use Cases for Smart City Projects

5.1 Smart Waste Management

Waste management is a topic that every modern or smart city needs to deal with [10] because it's necessary as well as of utmost importance for the health and hygiene of every citizen. The process in itself consists of many phases and to automate these processes is a goal of the waste management system for every municipality corporations of a proposed smart city [45]. There are various phases in the process such as collecting the waste, transporting it to a proper place, disposal of the same while taking care about the type of waste materials. This in-turn incurs a lot of expense, monitoring and maintenance while performed manually and also has to be completed on time so as to avoid spread of diseases across the city. The recycling process of the waste is also to be improved and needs to speed up [46]

5.2 IoT-Based Smart Waste Management Systems

The system has many stakeholders which are involved at various levels of the whole process and coordination amongst them needs to be sound in order to finish the work in optimum timeline. IoT has been observed as a potential solution to automate the process in order to make the process smart with the use of sensors and actuators [45].

Table 2 Comparison of various approaches for smart city projects with fog deployment models

Approach	Methodology	Merits	Limitations
Dynamic Module Deployment in a Fog Computing Platform [36]	Open-source platforms and virtualization technologies such as Kubernetes, OpenStack, and SaltStack with Module Deployment Algorithm (MDA) are proposed	Near-optimality, Real-timeness, Quick deployment, High scalability	With the use of container as virtualization technology, the performance is better but, only when there are less number of devices
Smart Agents and Fog Computing for Smart City Applications [38]	Combination of fog computing and a distributed agent-based layer wrapped in Virtual Objects to hide heterogeneity and to cope up with real-time issues	Less data transfer towards remote hosts, Good performance due to local access and computation, Dynamic addition/removal of VOs	Only limited to few smart city application services
Cost-Efficient Deployment of Fog Computing Systems at Logistics Centers in Industry 4.0 [39]	Proposes the deployment of an intelligent computing system with an integer programming model for a logistics center	Minimal installation cost, Increased computational efficiency due to the hybrid algorithm Discrete Monkey Genetic Algorithm (DMGA)	Limited to moderate-sized logistic centers due to incapability of dynamic deployment of fog devices
Docker Container Deployment in Fog Computing Infrastructures [40]	Uses container-based infrastructures for deployment & proposes optimizations which reduce the container deployment time	Quick deployment that speeds up the process by 4 times, Eliminates unnecessary delays	Swift and efficient deployment is held in large-scale infrastructures only
Fogernetes: Deployment and Management of Fog Computing Application [41]	Proposes a labeling system using Kubernetes that allows defining capabilities of fog nodes which maps them with the requirements of application’s components	Manages deployment of fog applications with heterogeneous devices. Ensures quality of service, availability, and reliability	Deployment evaluation was executed inside an isolated network, so weak network may impact performance and functionality

(continued)

Table 2 (continued)

Approach	Methodology	Merits	Limitations
Foggy: A Framework for Continuous Automated IoT Application Deployment in Fog Computing [42]	Presented a framework that automatically deploys the IoT application that provides updation to optimize resources in heterogeneous environments of IoT	Privacy preservation, Dynamic workload, Heterogeneity of IoT devices	Scalability and policy-based resource management process with optimization techniques is still at basic level
Deployment of Multi-tier Fog Computing System for IoT Services in Smart City [43]	Proposes a mathematical model for the multi-tier network of fog nodes by applying weighted sum algorithm and meta-heuristic algorithms	Cost optimization in fog network design, Reduced data traffic	Proposed model performs well in test scenario but yet to be evaluated in real-world scenario
Deployment of IoT Edge and Fog Computing Technologies to Develop Smart Building Services [44]	Proposes architecture of edge and fog layers. Fog nodes uses a cloud platform & MQTT (Message Queuing Telemetry Transport) is experimented as a protocol for communication	Improved interoperability and service scalability	Improvements in optimization is still required

This involves setting up a network of sensors and actuators, placed at different points of the city in order to collect data. The network is backed by cloud computing which enables inter communication between those devices for transferring the data [47]. To make sure that there is minimum latency and overhead in the whole process, fog computing is also introduced. Pooja et al. [46] suggests the deployment of required sensors on garbage bins which notifies the garbage truck drivers on their smartphones to empty the bin when the bin is full. This notification is gone via fog node which is connected to bins as well as with cloud, along with the notification, the garbage truck drivers also receive the shortest path from the fog node.

Al-Masri et al. [48] proposed an architecture where the cameras are attached to garbage bins which capture the images of disposed waste. These cameras are connected to the edge devices which process the images locally at the edge of the network for violation of disposed waste materials. This data is then, forwarded to the cloud for more analysis and monitoring. Hence, at the source level itself, it is possible to identify the violations with the help of machine learning algorithms without any help from the cloud. This reduces network traffic as less data is transmitted towards the cloud and real-time processing of images is done at the local level instead of sending it to a higher level (cloud), which can cause several delays. The approach also saves from many computational costs associated with conventional IoT architecture without local processing.

5.3 Proposed Fog Computing Based Smart Waste Management Architecture

The proposed architecture of "Smart Waste Management", is presented is shown in Fig. 6. The architecture proposed here consists of different components and the whole process related to smart waste management starts with "Trash Bins". Trash Bins located at main areas of the city are equipped with sensors, that are connected through the IoT network architecture with other components. When the bin is full or exceeds the threshold capacity, a notification is sent to the nearest "Fog Node". The fog nodes are installed together along with the city's existing video surveillance camera setup, with the intention of maintaining a single setup area and making sure that sufficient distance is maintained between the fog nodes (i.e. Video Surveillance Camera setup is chosen just as a mapping point for the location of fog nodes). The fog nodes are connected with the "Cloud Node" for extensive storage, processing, and analytics purpose. The cloud nodes share the important updates and notifications to the concerned city departments such as Waste Management, Waste Re-cycle and Healthcare. Suppose, Trash Bin(1)-Connected with Fog Node 1, is full, then a notification with location of the trash bin is sent via fog node to the nearest "Trash Collector" on the smartphone (For the sake of simplicity, only one trash bin is connected to the nearest fog node, in reality, there can be more than one). Once the trash bin is

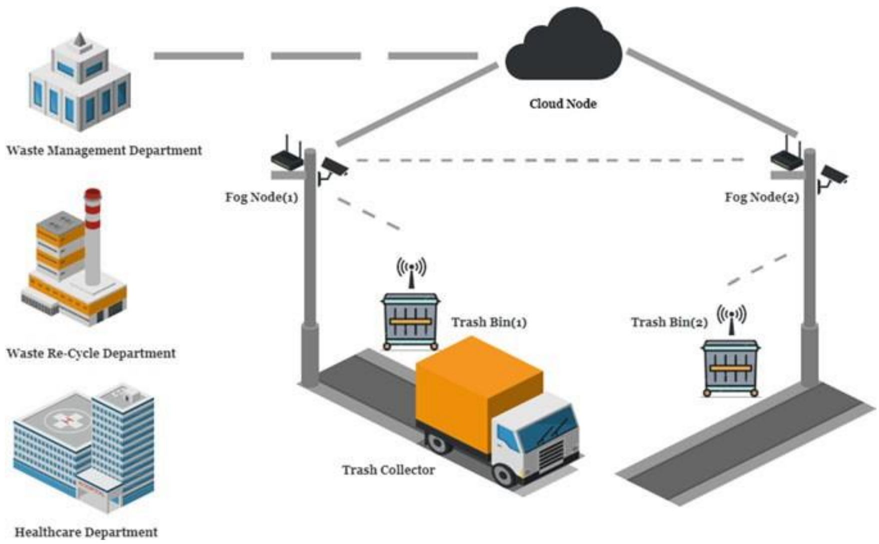


Fig. 6 Smart waste management architecture

emptied by the trash collector, an update would be sent to the same fog node through the sensors installed with the trash bin.

5.4 *Execution Flow of Proposed Smart Waste Management Architecture*

Now, if Trash Bin(2)-Connected with Fog Node 2 is full, not allocated to any other trash collector and is in the nearest range of Fog Node 1 then a communication would be done between Fog Node 1 and Fog Node 2 and the trash bin would be allocated to the same trash collector. Hence, there would be continuous communication between all the fog nodes for the current status of the trash bins. In this architecture, due to the inclusion of fog computing, the latency would be minimal, scalability is high as we can install or remove the fog nodes dynamically as per our requirement. The process is happening in real-time that leads to good performance. A collective update from all the fog nodes on the trash bin status, is shared to the cloud node periodically. This update is then stored, processed, and forwarded to the concerned city departments for monitoring of the whole process throughout the day. The monitoring is aided through edge analytics performed by the fog nodes on the trash bin data and the important notifications observations are sent towards the cloud. (For example, Trash bin in some area not being emptied since X number of days or Trash bin in some area getting full frequently due to crowded population in that area).

5.5 Smart Parking

With the rise of people moving into cities from towns and villages, there has also been raised in the number of vehicles in the cities. The high quality of living and daily commuting to the workplace, also adds to the contribution of increased vehicular traffic. The common practice of searching a parking space in any urban city is usually through finding a space on the street. The process in itself is time-consuming and also is difficult to follow in a urban city with high vehicle density [49]. Hence, finding a perfect location for the parking vehicles in a city is a big challenge these days which ultimately wastes the time and fuel [50]. This issue can be resolved by introducing IoT, where a smart parking system is developed, supported by ad-hoc networks and road-side sensors/actuators i.e. Vehicular ad-hoc Networks (VANET's) [51]. Though, only VANET based parking systems face several challenges such as data collection and data sharing with other vehicles by using the ad-hoc network which cannot be relied upon when timely updates are required [52]. Hence, Fog computing architecture fits here as all the needed resources would be available at the edge of the network. Fog nodes introduced at closer proximity of parking premises will give information about available parking space to the vehicles, helping them to park their vehicles [51]. These fog nodes have the potential of storing and processing the data as well as communicating with different fog nodes deployed near other parking premises. Well, more efficiency and optimization can be tuned in by the help of remote cloud computing as well.

5.6 IoT Based Smart Parking Architectures

Different layers are defined in the architecture proposed by Celaya-Echarri et al. [53]. The IoT layer includes parking sensors and all these sensors would be communicating in a mesh network for data exchange. The vehicles and the parking sensors interact with each other as well as with the fog nodes in the fog layer. The fog node consists of three components namely, communication interface, control subsystem, and a beacon interface, where communication transceiver helps in exchanging data between parking sensor nodes, beacon interface sends the beacons to vehicles for sharing data without the need to establish a connection. This connectivity gives speed to the process when compared to the conventional connection-oriented data transmission. The control subsystem manages and provides fog services to parking sensors and it also communicates with the cloud through a gateway layer which performs data routing tasks.

Tandon et al. [52] have proposed a smart parking architecture in which the developed framework, the on-line registration for vehicles parking amenities has been given. Users can apply for the reservation of car parking online. Once the car owner has completed a one-time virtual identity verification, they will secure the parking location before approaching the spot. One can even revoke a parking appointment if there is a prior appointment made for the parking area. In this case, the person in the pending list would be assigned room in the parking area. The online reservations details are then modified to the cloud such that the position in the designated parking area can be assigned to other car owners who want the parking area. Each parking details must be matched with the parking model of the fog system for intelligent parking purposes. The virtual procedure for car parking reservations is accompanied by the use of fog nodes. Fog nodes regularly modify the accessibility of cloud parking space. The driver of the car, whose parking spot is assigned prior, must arrive at the parking area in good time or the parking area would be assigned to someone on the waiting list. On-line car registration parking service is more advantageous than the on-site parking system currently in operation in most places. Online appointment parking can be reserved for a car park on almost any given day. Once the position has been allocated to the car, the cumulative expense of the vehicle is determined by taking into account the waiting period and the period of the time.

Zhang et al. [54] have proposed an architecture Smart VFC (Vehicular Fog Computing) system, integrating both PVA (Parked Vehicle Assistance) and smart parking. A VFC aware intelligent parking bid is introduced to direct vehicles on-the-go to the available parking spaces with less intervention. To leverage the fog potential of parked vehicles to facilitate delay-sensitive computing capabilities with cash incentives to reimburse for their overhead expenses. The recommended assignment rule significantly increases the overall utility of smart vehicles and the revised payout policy assures reward flexibility, human efficiency, and budgetary harmony. They also have an analysis stage with adaptive offload pricing changes to increase the performance of offloading and the benefit of the fog method.

Kamran et al. [55] have proposed an architecture in which the architecture suggested comprises 3 levels. The first layer includes the camera over the parking channels or spaces and is able to record the photographs of the available parking and decide whether or not the parking slot is filled. The second layer of the network is a fog node that is attached to the cameras via a microcontroller interface. Fog nodes handle photographs of parking slots, and details on parking slots are seen on the LED. When there is no available parking space in a specific parking area, the details on the nearby accessible parking were shown on the relevant LED parking. The third layer includes a cloud that is attached to fog and is liable for preserving and handling visual information for extensive amounts of time When the driver enters the parking spot, the LED panel shows the state of the parking spaces to the driver. The suggested system would not require a mobile to check details on empty parking spaces. Therefore, in order to reduce traffic chaos in parking spots and fuel emissions, drivers are routed into the parking area at the main door of the parking lot. Simulations are extensively performed to test the feasibility and robustness of the emerging fog-enabled car parking structure. Interdisciplinary assessments indicate a substantial decrease in congestion and network use relative to the cloud-based version of the intelligent parking scheme. Through use of cameras to monitor parking space is a drawback of the proposed study. This will give rise to privacy concerns for car owners when the pictures are processed in the cloud. With an enhanced quantity of car park areas, adjusting the load on fog nodes will indeed be necessary to sustain efficiency.

5.7 Proposed Fog Computing Based Smart Parking Architecture

The proposed architecture of “Smart Parking”, is presented is shown in Fig. 7. The architecture proposed here, has to be implemented with the inclusion of Video Surveillance Cameras installed at different parking zones, “Cam(1) and Cam(2)”. Fog Node(1) are Fog Node(2) are connected with Cam(1) and Cam(2) respectively. Users who want to park their vehicles have to pre-book their parking place through a smartphone application. The smartphone initially, sends the parking request notification to the nearest fog node from the user (source). Then, that fog node communicates with the nearest fog node from the destination regarding the availability of the parking space (which is known with the help of video surveillance camera’s feed). The fog nodes keep updating themselves with the parking space status continuously, so as to align with the incoming parking requests.

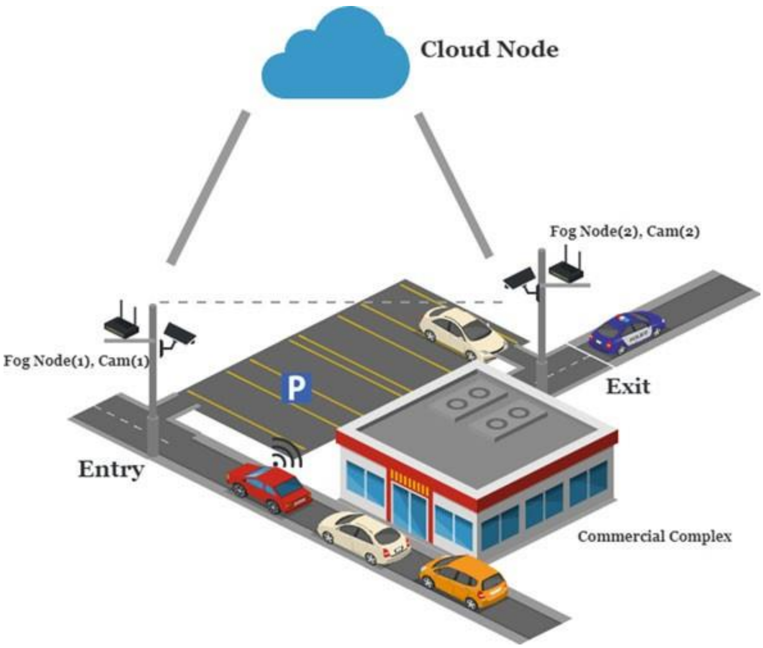


Fig. 7 Smart parking architecture

5.8 Execution Flow of Proposed Architecture

For example, user can only pre-book the parking space when he/she is 1 mile nearer to the destination and also has to enter the "Vehicle Number" while booking the parking space. A user is 0.5 miles away from destination, books a parking space at 5.00 PM from a smartphone application. The application suggests the "Estimated Time" to reach the destination from source which is calculated on 2 factors, (1) Google Maps Distance Time (2) 10 min of buffer. If the estimated time to reach the destination (suggested by Maps) is 5.15 PM, then the parking space would be blocked till 5.25 PM (5.15+10 min buffer). If the parking space is available, then the parking space ID would be sent directly to the user else the user would be listed in the waiting list. When the user arrives to the parking area, Fog Node(1) and Cam(1) comes into action by detecting the vehicle's number plate and comparing it with the vehicle numbers registered within the same booking time. That object detection and processing task is handled together by Fog Node(1) and Cam(1) and after the parking is done, the parking space status is updated. After parking successfully on the parking space ID and at the time of exiting, Fog Node(2) and Cam(2) would detect the vehicle's number plate and update the parking space status (If there's any request in the waiting list, they would be assigned the space).

This architecture serves to be working in real-time and helps to maintain the flow of parking for users as the conventional way of parking is tedious and time-consuming. The introduction of fog computing makes the task quick as well as efficient. The periodic and collective update would be sent to the Cloud Node regarding the parking space status and analytics revolving around that data. For example, there can be a peak hour on weekends when parking space is occupied most of the time and the time between vehicle's entry and exit is longer than the usual.

6 Conclusion

In a general prospect, the basic research findings of "Smart City" as a concept which is to be implemented in real-time, brings along many challenges. To an extent, these challenges can be overcomed by the exploiting the benefits of Fog Computing. Unreliable latency, transmission overhead, and slower data processing are few paradigms where fog computing has shown its potential. Along with the fog computing, the integration of automation as well as data exchange trends of Industry 4.0, aids to a better growth in the smart city projects. Our research work presents a methodology for deploying fog computing architecture in applications such as Smart Waste Management and Smart Parking. The proposed architectures would offer better and efficient services in contrast to the conventional approaches.

7 Future Work

Future research work would be in the direction of prototyping applications of the proposed research study. The technical procedures and aspects of fog architecture deployment for Smart Waste Management and Smart Parking would be given prime importance.

References

1. Petrolo R, Loscri V, Mitton N (2017) Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms. Trans Emerg Telecommun Technol 28(1):e2931
2. Ejaz W, Naeem M, Shahid A, Anpalagan A, Jo M (2017) Efficient energy management for the internet of things in smart cities. IEEE Commun Mag 55(1):84–91
3. The top 10 IoT segments in 2018 based on 1600 real IoT projects, Feb 2018
4. The top 10 IoT application areas based on real IoT projects, Aug 2016.
5. Rathore MM, Ahmad A, Paul A, Rho S (2016) Urban planning and building smart cities based on the internet of things using big data analytics. Compu Netw 101:63–80
6. Gluhak A, Krco S, Nati M, Pfisterer D, Mitton N, Razafindralambo T (2011) A survey on facilities for experimental internet of things research. IEEE Commun Mag 49(11):58–67
7. Alavi AH, Jiao P, Buttlar WG, Lajnef N (2018) Internet of things-enabled smart cities: State-of-the-art and future trends. Measurement 129:589–606
8. PricewaterhouseCoopers. Creating the smart cities of the future
9. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M (2015) Internet of things: a survey on enabling technologies, protocols, and applications. IEEE Commun Surv Tutorials 17(4):2347–2376
10. Perera C, Qin Y, Estrella JC, Reiff-Marganiec S, Vasilakos AV (2017) Fog computing for sustainable smart cities: a survey. ACM Comput Surveys (CSUR) 50(3):1–43
11. Talari S, Shafie-Khah M, Siano P, Loia V, Tommasetti A, Catalão JPS (2017) A review of smart cities based on the internet of things concept. Energies 10(4):421
12. Zanella A, Bui N, Castellani A, Vangelista L, Zorzi M (2014) Internet of things for smart cities. IEEE Internet Things J 1(1):22–32
13. Atlam HF, Walters RJ, Wills GB (2018) Fog computing and the internet of things: a review. Big Data Cogn Comput 2(2):10
14. Sharma I, Tiwari R, Anand A (2017) Open source big data analytics technique. In: Proceedings of the international conference on data engineering and communication technology. Springer, pp 593–602
15. Vyas T, Desai S, Ruparelia A (2020) Fog data processing and analytics for health care-based IoT applications. In: Fog data analytics for IoT applications. Springer, Berlin, pp 445–469
16. Yi S, Li C, Li Q (2015) A survey of fog computing: concepts, applications and issues. In: Proceedings of the 2015 workshop on mobile big data, pp 37–42
17. Yi S, Hao Z, Qin Z, Li Q (2015) Fog computing: Platform and applications. In 2015 Third IEEE workshop on hot topics in web systems and technologies (HotWeb). IEEE, pp 73–78
18. Ning Z, Huang J, Wang X (2019) Vehicular fog computing: enabling real-time traffic management for smart cities. IEEE Wirel Commun 26(1):87–93
19. Zahmatkesh H, Al-Turjman F (2020) Fog computing for sustainable smart cities in the IoT era: caching techniques and enabling technologies—an overview. Sustain Cities Soc 59:102139
20. Zhang C (2020) Design and application of fog computing and internet of things service platform for smart city. Futur Gener Comput Syst 112:630–640

21. Lin SY, Du Y, Ko PC, Wu TJ, Ho PT, Sivakumar V, Subbareddy R (2020) Fog computing based hybrid deep learning framework in effective inspection system for smart manufacturing. Comput Commun 160:636–642
22. Naranjo PGV, Pooranian Z, Shojafar M, Conti M, Buyya R (2019) Focan: a fog-supported smart city network architecture for management of applications in the internet of everything environments. J Parallel Distrib Comput 132:274–283
23. Nasir M, Muhammad K, Lloret J, Sangaiah AK, Sajjad M (2019) Fog computing enabled cost-effective distributed summarization of surveillance videos for smart cities. J Parallel Distrib Comput 126:161–170
24. Srinivasa Desikan KE, Kotagi VJ, Siva Ram Murthy C (2020) Topology control in fog computing enabled iot networks for smart cities. Comput Netw 176:107270
25. Kumar NM, Goel S, Mallick PK (2018) Smart cities in India: features, policies, current status, and challenges. In: 2018 technologies for smart-city energy security and power (ICSESP). IEEE, pp 1–4. IEEE
26. Gupta K, Hall RP (2017) The Indian perspective of smart cities. In: 2017 smart city symposium Prague (SCSP). IEEE, pp 1–6
27. Prasad D, Alizadeh T (2020) What makes Indian cities smart? A policy analysis of smart cities mission. Telematics Inform 55:101466
28. Chatterjee S, Kar AK, Gupta MP (2018) Success of IoT in smart cities of India: an empirical analysis. Govern Inf Q 35(3):349–361
29. Jawaid MF, Khan SA (2015) Evaluating the need for smart cities in India. Int J Adv Res Sci Eng IJARSE 4(01)
30. Lasi H, Fettke P, Kemper HG, Feld T, Hoffmann M (2014) Industry 4.0. Bus Inf Syst Eng 6(4):239–242
31. Chauhan A, Jakhar SK, Chauhan C (2020) The interplay of circular economy with industry 4.0 enabled smart city drivers of healthcare waste disposal. J Cleaner Product 279:123854
32. Kumar S, Tiwari R (2021) Dynamic popularity window and distance-based efficient caching for fast content delivery applications in CCN. Eng Sci Technol Int J
33. Dalenogare LS, Benitez GB, Ayala NF, Frank AG (2018) The expected contribution of industry 4.0 technologies for industrial performance. Int J Product Econ 204:383–394
34. Lom M, Pribyl O, Svitek M (2016) Industry 4.0 as a part of smart cities. In: 2016 smart cities symposium Prague (SCSP). IEEE, pp 1–6
35. Safiullin A, Krasnyuk L, Kapelyuk Z (2019) Integration of industry 4.0 technologies for âeœsmart citiesâe development. In: IOP conference series: materials science and engineering, vol 497. IOP Publishing, pp 012089
36. Hong HJ, Tsai PH, Hsu CH (2016) Dynamic module deployment in a fog computing platform. In: 2016 18th Asia-pacific network operations and management symposium (APNOMS). IEEE, pp 1–6
37. Bonomi F, Milito R, Zhu J, Addepalli S (2012). Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on mobile cloud computing, pp 13–16
38. Giordano A, Spezzano G, Vinci A (2016) Smart agents and fog computing for smart city applications. In: International conference on smart cities. Springer, Berlin, pp 137–146
39. Lin CC, Yang JW (2018) Cost-efficient deployment of fog computing systems at logistics centers in industry 4.0. IEEE Trans Indus Inform 14(10):4603–4611
40. Ahmed A, Pierre G (2018) Docker container deployment in fog computing infrastructures. In: 2018 IEEE international conference on edge computing (EDGE). IEEE, pp 1–8
41. Wöbker C, Seitz A, Mueller H, Bruegge B (2018) Fogernetes: deployment and management of fog computing applications. In: NOMS 2018–2018 IEEE/IFIP network operations and management symposium. IEEE, pp 1–7
42. Yigitoglu E, Mohamed M, Liu L, Ludwig H (2017) Foggy: a framework for continuous automated IoT application deployment in fog computing. In: 2017 IEEE international conference on AI & mobile services (AIMS). IEEE, pp 38–45

43. Maiti P, Apat HK, Kumar A, Sahoo B, Turuk AK (2019) Deployment of multi-tier fog computing system for IoT services in smart city. In: 2019 IEEE international conference on advanced networks and telecommunications systems (ANTS). IEEE, pp 1–6
44. Ferrández-Pastor F-J, Mora H, Jimeno-Morenilla A, Volckaert B (2018) Deployment of IoT edge and fog computing technologies to develop smart building services. Sustainability 10(11):3832
45. Shukla ROLI (2016) Smart waste management. Asian J Pharm Educ Res 5(2):48–54
46. Garach PV, Thakkar R (2017) A survey on fog computing for smart waste management system. In: 2017 international conference on intelligent communication and computational techniques (ICCT). IEEE, pp 272–278
47. Aazam M, St-Hilaire M, Lung CH, Lambadaris I (2016) Cloud-based smart waste management for smart cities. In: 2016 IEEE 21st international workshop on computer aided modelling and design of communication links and networks (CAMAD). IEEE, pp 188–193
48. Al-Masri E, Diabate I, Jain R, Lam MH, Nathala SR (2018) Recycle. Io: an IoT-enabled framework for urban waste management. In: 2018 IEEE international conference on big data (big data). IEEE, pp 5285–5287
49. Tiwari R, Sharma HK, Upadhyay S, Sachan S, Sharma A (2019) Automated parking system-cloud and IoT based technique. Int J Eng Adv Technol (IJEAT) 8(4C):116–123
50. Aydin I, Karakose M, Karakose E (2017) A navigation and reservation based smart parking platform using genetic optimization for smart cities. In: 2017 5th international Istanbul smart grid and cities congress and fair (ICSG). IEEE, pp 120–124
51. Tang C, Wei X, Zhu C, Chen W, Rodrigues JJPC (2018) Towards smart parking based on fog computing. IEEE Access 6:70172–70185
52. Tandon R, Gupta PK (2019) Optimizing smart parking system by using fog computing. In: International conference on advances in computing and data sciences. Springer, pp 724–737
53. Celaya-Echarri M, Iván Froiz I, Azpilicueta L, Fraga-Lamas P, Lopez-Iturri P, Falcone F, Fernández-Caramés TM (2020) Building decentralized fog computing-based smart parking systems: from deterministic propagation modeling to practical deployment. IEEE Access 8:117666–117688
54. Zhang Yi, Wang C-Y, Wei H-Y (2019) Parking reservation auction for parked vehicle assistance in vehicular fog computing. IEEE Trans Veh Technol 68(4):3126–3139
55. Awaisi KS, Abbas A, Zareei M, Khattak HA, Khan MUS, Ali M, Din IU, Shah S (2019) Towards a fog enabled efficient car parking architecture. IEEE Access, 7:159100–159111

Integration of Fog Computing and IoT-Based Energy Harvesting (EHIoT) Model for Wireless Sensor Network

H. M. K. K. M. B. Herath, R. D. D. Prematilake, and B. G. D. A. Madhusanka

Abstract The evolution of IoT-based Energy Internet (EI) applications has been discussed in a detailed study of modern IoT (Internet of Things) technologies for smart grids and smart urban ecosystems (SUE), such as smart cities, smart buildings, smart metering, and energy storage infrastructures. Fog computing is a revolutionary networking architecture that enables edge nodes to share, compute and store data resources. One of the most compelling scenarios in IoT is energy consumption while communicating information through a wireless sensor network (WSN). In this scenario, sensors, actuators, and smart devices communicate and consume a large amount of energy. Since Fog nodes are widely spread and mostly operate on batteries, effective energy storage should be considered to extend the network lifespan. In this chapter, we proposed a design model for the integration of a wireless sensor network for the hospital environment by enabling IoT-based energy harvesting (EHIoT) methods. This approach combines collecting healthcare-related data from the hospital environment and monitoring them through the IoT/Fog-based system. We applied EHIoT methodologies for harvesting energy and managing them in the proposed WSN. Ambient-based and human-based energy harvesting methods were used to design the proposed EHIoT system.

Keywords EHIoT · Energy Internet (EI) · Fog computing · Internet of Things (IoT) · Smart healthcare · Wireless sensor network (WSN)

H. M. K. K. M. B. Herath (✉) · B. G. D. A. Madhusanka
Faculty of Engineering Technology, The Open University of Sri Lanka, Nugegoda, Sri Lanka

B. G. D. A. Madhusanka
e-mail: bgmad@ou.ac.lk

R. D. D. Prematilake
Faculty of Science and Technology, University of Plymouth, Plymouth, UK

R. Tiwari et al. (eds.), *Energy Conservation Solutions for Fog-Edge Computing Paradigms*, Lecture Notes on Data Engineering and Communications Technologies 74,
https://doi.org/10.1007/978-981-16-3448-2_11

1 Introduction

We are experiencing the beginning of a new revolution called the Internet of Things (IoT) or the Internet of Objects. The smart city idea has recently been made available in a range of implementations where smart communication seeks to increase the quality of life by reducing costs [1, 2]. The implementation of the Internet of Things (IoT), urban IoT, and the Energy Internet (EI) based applications are mainly focused on public utilities such as smart transportation and regulatory systems, emergency medical services (EMS), healthcare services, and institutional health monitoring networks [3–5]. IoT is one of the new networking methodologies that include a more secure and effective connectivity process, consisting of data collection, encoding, delivery, and storage phases. Unlike other networking platforms, the IoT platform has several advantages, for instance, the ability to deliver several complex communication scenarios in real time. Another example would be reducing the use of hardware would make the use of machines more effective. In the context of modern wireless telecommunications, the IoT is a new concept that is rapidly creating advances. In general, IoT relates to the networked interconnection of ordinary things, often packed with pervasive knowledge. It also opens up vast possibilities for a broad range of new technologies to increase the standard of human life due to exponential advances in the underlying technology. In recent years, IoT has significantly attracted attention from academics and clinicians all across the globe. The IoT layer primarily consists of physical, network, and application layers [6]. The Internet Engineering Task Force (IETF) has established alternative protocols for the contact between IoT [7] devices using IP because of the stable and secure standard of IP [8, 9]. Wi-Fi, Bluetooth, LoRa, Zigbee, Z-Wave, NFC, and Sigfox are typical IoT networks used in smart health models. Table 1 depicts the features of the different IoT networks used in modern smart healthcare applications.

The Internet of Things (IoT) links a wide range of nodes, including not only machines such as servers and clients, but also devices such as sensors and actuators [10]. Wireless sensor networks (WSNs) [11, 12] are innovative technology in medical physics that could change human lifestyles. Wireless sensors are the smallest network devices with special features such as support for large-scale deployment, flexibility, and reliability. The healthcare sector has reached dramatic development in the twenty-first century thanks to the wireless medical sensor networks (WMSNs) in modern healthcare applications. Sensor data is transferred to cloud servers [13] and analyzed to determine the action to be performed by the actuators. An intermediate layer called the Fog layer [10] is added to the IoT between devices and clouds. Fog computing is strongly virtualized and offers a link between end devices and the cloud for computing, storage, and networking [14]. The basic principle of Fog computing is the relocation of data center activities to Fog nodes on the edge of the network. In this section, we proposed a design model of a Fog network for the WSN in the hospital environment.

Another significant factor is the wellness and well-being of the smart city model [15]. Health monitoring systems in smart environments are attracting massive growth

Table 1 Comparison of different IoT network protocols

Criteria	Wi-Fi	Bluetooth	LoRa	Zigbee	Z-Wave	Cellulaire	NFC	Sigfox
Specification	Based on 802.11n	Bluetooth 4.2 core specification	LoRaWAN	ZigBee 3.0 based on IEEE802.15.4	Z-Wave Alliance ZAD12837/ITU-T G.995	GSM/GPRS/EDGE (2G), UMTS/HSPA (3G), LTE (4G)	SO/IEC 18000–3	Sigfox
Network type	LANWPAN/P2P	LAN	LAN	LAN	LAN	MAN	P2P	LPWAN
Topology	Star	Star	Star & Star of Star	Mesh, Star, Tree	Mesh	Mesh	Mesh	Star
Power	Low–high	Low	Very low	Very low	Very low	High	Very low	Low
Data rate	Up to 1.3 Gbps	2.1 Gbps	0.3 Gbps to 100 kbps	250 Gbps	40 Gbps	Up to 1Gbps	424 kbits	10–1000 bps
Frequency band	54 Mb/s	2.4 GHz	Various	868/915 MHz, 2.4 GHz	900 MHz (ISM)	850/900; 1800/1900 MHz	13.56 MHz	900 MHz
Range	Up to 100 m	<100 m	3–5 km urban area	10–20 m	30 km	Where signal reach	5 cm	30–50 km
Energy needed	–	High	–	Medium	–	–	–	–
Cost	Low	Low	Low	Good	Effective	Less and vice versa	Low	Relatively low
Security	WPA and WPA2	Shared secret	Per-device AES128 keys	CBC-MAC (ext. of CCM)	Data encryption	–	–	–
Applications	Any device with cellular connectivity	Network for data exchange headset	Smart city, sensor networks, industrial automation application	Senor networks Industrial automation	Residential lighting and automation	Use in Wi-Fi, ADSL, broadband, digital TV and Radio	Payment transactions, business transactions, contextual information	Smart grid

and provide complementary alternatives to conventional health care services. Modern machines can conveniently access a wide range of patient health records. Patient data will be spread in various locations due to technological advances on the Internet of Things and the Internet of Medical Things (IoMT). Cloud storage servers have been used in multiple healthcare surveillance networks to store and process many of the data collected from sensor nodes [16]. Consequently, scientists have suggested ways to manage data on healthcare dependent on cloud computing [17]. Vital data collected by IoT devices are typically processed [18] and we need to address the energy drawbacks in order to design real-time health care applications [19]. Energy harvesting (EH) is a process of converting ambient-based (solar, wind, etc.) and human-based (breath, motion, etc.) energy into electrical energy [20]. Power generation plays a key role in pushing IoT devices in an EH framework, which has several concurrent operating subsystems. Efficient energy storage for low-power applications has been achieved from several sources during the last decade.

Energy efficiency is one of the most important requirements of everyday living. The notion of energy-efficient appliances has emerged in various fields, such as lighting, telecommunications, healthcare, and so on. A significant method for assessing the energy efficiency of different equipment is energy management. Higher healthcare expenses are one of the biggest issues faced by governments, healthcare institutions, and relevant professionals [21]. In this chapter, we suggested a design model for the integration of the IoT/Fog, and wireless sensor network for the hospital environment (WSNHE) by enabling IoT-based energy harvesting (EHIoT) methods. The remainder of this chapter is as follows; Sect. 2 outlines the theoretical background of medical sensors, IoT/Fog computing in the healthcare sector, and energy use models of IoT/Fog and WSN. Section 3 addresses the related work. The proposed design model is presented in Sect. 4 and, finally, the proposed design is concluded in Sect. 5.

2 Background

The impact of IoT and Fog computing on the smart healthcare industry, as well as the energy consumption models used in IoT/Fog and wireless sensor networks (WSN), are briefly discussed in this section.

2.1 Medical Sensors

The sensors used in the medical sector for the detection, monitoring or treatments of diseases are classified as medical sensors. The application of sensors in the field of medicine and public health has a long tradition. Sensor networks for many health-related applications include offering interfaces for automated patient management, diagnostics, hospital medication administration, telemonitoring of human physiological data, and surveillance and monitoring of physicians and/or patients inside the hospital [22, 23]. Medical sensors are combined with transducers and signal processing algorithms to identify electrical, thermal, optical, biological, genetic, and other physiological signals of origin in order to determine the health condition of an individual. Early warning is necessary; disease or patient monitoring is vital to human survival in acute circumstances. Table 2 shows the various medical sensors available to measure or identify a patient's illness.

Wireless body sensor networks or also known as WBSNs are composed of two types of sensor nodes (sensing node and stimulating node) for medical and healthcare applications. The sensing nodes gather data, analyze, store, and send it over a wireless network. When medical attention is needed, the stimulating nodes are usually used. A sensor node comprises three sections: power, analogue, and digital. The analogue part of a multi-sense module, such as the sensors, is normally placed in a sensor node, while the wireless transceiver is placed in a transceiver module. The wireless transceiver and multi-sensor board are usually connected to the processor in the digital blocks via serial communication [24].

Table 2 Different medical sensors used in healthcare sector

Sensors	Description
Temperature probes	Used for measuring body temperature. It aims to get quality medication and health care. They're called thermometers as well
Force sensors	Used in kidney dialysis machines
Airflow sensors	Used in anesthesia delivery systems, laparoscopy, heart pumps
Pressure sensors	Used in infusion pumps and sleep apnea machines. Most of the pressure sensors are integrated with embedded systems. They are used for medical diagnosis, blood pressure monitoring, infusion pumps etc.
Implantable pacemaker	It is an embedded sensor device in real-time that provides the heart muscle with a synchronized rhythmic electrical signal to maintain an effective cardiac rhythm
Oximeter	It determines the fraction of hemoglobin filled with oxygen relative to the overall blood count of hemoglobin
Glucometer	Measures approximate blood glucose concentration
Magnetometer	Specifies the position of the device by observing the variations in the Earth's magnetic field surrounding the object
Electrocardiogram sensor	Tests the electrical function of the organ. It's called an ECG sensor
Heart rate sensor	Counts the amount of cardiac contractions per minute
Electroencephalogram sensor	Tests the electrical function of the brain
Electromyogram sensor	It tracks the electrical function of the skeletal muscles
Respiration rate sensor	It counts how many times your chest gets up in a minute

2.2 IoT and Fog Computing in Smart Healthcare

Fog computing is heavily virtualized, providing connectivity, storage, and networking synchronization between end devices and the cloud [14]. The transition of data center processes to Fog nodes on the network's edge is the core principle of Fog computing. Fog nodes are located between the device's layers and the cloud. Fog nodes in the network are linked to one another and the IoT connects a wide variety of electronics, including sensors in addition to computers. The cloud computing model network sends sensor data to cloud servers, where it is analyzed. Networks are overburdened and processors are swamped with sensor data. Data and processes are spread not only to cloud servers but also to Fog nodes in Fog computing models to minimize waiting time and network traffic while maximizing system reliability. Even though it is possible to minimize server traffic in a cloud, the overall electrical energy used by Fog nodes is increased to process sensor data. There are two types of nodes that can be used by devices:

(1) Sensor nodes: A sensor node collects sensor data from the real world, such as temperature, and sends it to one or more of the fog layer's edge nodes.
(2) Actuator nodes: On receiving intervention from an edge node, an actuator node performs the real physical procedures.

Fog nodes are intertwined with each other in the networks. A Fog node collects sensor node data as well as other Fog nodes data. Information collected by the Fog node is transferred to nearby Fog nodes and cloud servers. In addition, based on sensor data obtained from sensor nodes and Fog nodes, a Fog node takes a decision on what action sensor nodes are going to performed. All sensor nodes' data were sent to the servers of the conventional cloud computing system. The sensor data will then be processed and the servers will produce the processed data. The sensor information and the expertise given during the sensor data processing are stored in the databases of the server. The servers transmit actions to the actuator nodes in the network.

2.3 Energy Consumption Models

In this section, we explain the energy consumption models for network devices. Network devices are divided into two types:

1. Equipment that are shared by many end-users.
2. Customer premises equipment (CPE) that is dedicated to a single user or few users.

To distribute the power usage across the data traffic movements, we propose a flow-based energy model dependent on highly shared equipment of the proposed wireless sensor network of the hospital environment. Sections 1 and 2 define the flow-based energy consumption model and energy usage model of the wireless sensor networks.

2.3.1 Flow-Based Energy Consumption Model

The energy usage measurement of the cloud infrastructure depends on the relative power distribution of the system's overall flow by devices used in different users and services, such as routers and switches in the center of the network. This is also referring to as the model of "flow-based" energy uses in the network [25–27]. Network equipment is consuming power, whether the equipment is inactive or working. The power usage of a regular network interface can be modeled in linear form as seen in Eq. (1). Figure 1 illustrates the power consumption trend versus load for standard network equipment.

$$E(C) = E_{idle} + C \cdot \frac{E_{max} - E_{idle}}{C_{max}} = E_{idle} + C\, E_b \tag{1}$$

where

- $E(C)$ Power consumption under load C
- C Load of a network equipment
- E_{idle} Idle power consumption of network equipment
- E_{max} Maximum power consumption of network equipment
- C_{max} Maximum capacity/load of network equipment.

The idle power (P_{idle}) can be a significant proportion of P_{max} (up to more than 90%), when measuring a service's energy consumption, P_{idle} cannot be ignored. Since the majority of network systems have a linear power profile [16], the same model is used for all network equipment shared by various users and services.

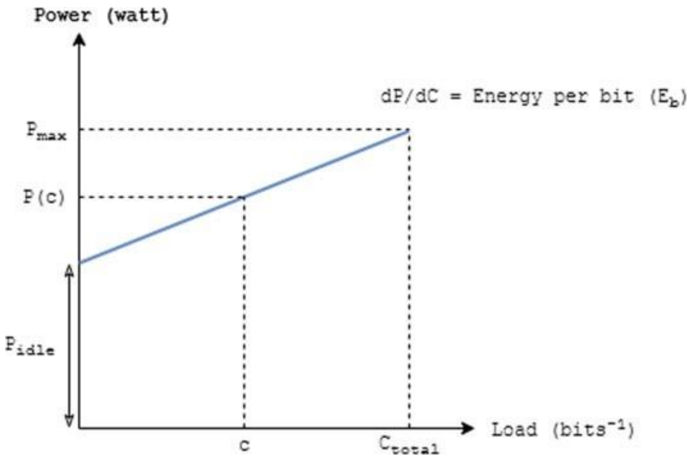


Fig. 1 Behavior of the power consumption trend versus load for network equipment

2.3.2 Energy Model of WSN

Processor, transceiver and sensor units are the three main units in a WSN-based system that consumes most of the system's resources [28]. The energy models for all three units are developed in this section.

(1) Energy consumption model of the processor unit: Active mode (AM) and sleep mode (SM) are the two modes in which the processor can run. The processor's energy consumption (E_{P0}) model is defined in Eq. (2).

$$E_{P0} = \sum_{j=1}^{N} \max(E_{AM}, E_{SM}) \tag{2}$$

(2) Energy consumption model of the WSN transceiver: The following equations describe the power consumption of the WSN transceiver. The transmitter or receiver will either be functioning in three phases knows as, idle, active (transmit (TX) or receive (RX)), or sleep. Equations 3 and 4 describe the energy consumption model at the transmitting (E_{T0}) and receiving (E_{R0}) stages. The maximum value of all three states to be constantly calculated will then be power consumption at any given time.

$$E_{T0} = \max\{E_T(d), E_{Tidle}, E_{Tsleep}\} \tag{3}$$

$$E_{R0} = \max\{E_R(d), E_{Ridle}, E_{Rsleep}\} \tag{4}$$

(3) Energy consumption model of the sensor unit: the overall number of nodes (j) and the number of sensors per node (k) are used in the energy model of the sensors. Equation (5) indicates the average power of all sensors. It summarizes the energy used by the sensors of each node.

$$E_{Sensor} = \sum_{j=1}^{N}\sum_{k=1}^{N_S}(E_{jk}) = \sum_{j=1}^{N}\sum_{k=1}^{N_S} V_{jk}(T + E_S)I_{jk}(T + T_S) \tag{5}$$

where

E_{P0}	Processor energy
E_{AM}	Processor active mode
E_{SM}	Processor sleep mode
N	Number of WSN nodes
E_{T0}	Transmitter energy
E_{R0}	Receiver energy

E_{Tidle}	Energy at transmitter "idle" state
E_{Ridle}	Energy at receiver "idle" state
E_{Tsleep}	Energy at transmitter "sleep" state
E_{Rsleep}	Energy at receiver "sleep" state
E_{Sensor}	Sensor energy
T	Instant of time
Ts	Sampling time
V_{jk}	Voltage of the kth sensor of jth node
I_{jk}	Current of the kth sensor of jth node

3 Related Works

By incorporating any object for interaction through embedded systems, IoT would increase the emergence of the Internet, leading to a highly distributed network of devices interacting with human beings as well as other devices. IoT opens up enormous opportunities for a vast variety of novel solutions that aim to enhance the quality of our lives, due to rapid developments in the underlying technology. Internet of Things architectures also faces many challenges with regard to power efficiency and scalability at low cost to achieve massive widespread deployment, especially in clinical environments. In recent years, researchers and practitioners from around the world have drawn a great deal of interest to the energy efficiency of healthcare-related IoT applications. This section introduces the previous studies in terms of Energy Internet, IoT, and Fog computing technologies. We were primarily interested in research related to energy harvesting techniques in the healthcare industry. The following research studies were discovered after a thorough search on Google Scholar.

New technology such as the Internet of Things (IoT) provides a wide variety of solutions in the field of energy. The IoT is used to boost energy performance, increase green energy contributions and reduce the environmental impact of energy use. Kabalci et al. [29] have thoroughly analyzed Internets of Things (IoT) systems for smart grids and intelligent ecosystems, like smart city, smart housing, smart metering, and energy storage technology to explore the development of EI-based IoT technologies. The Energy Internet (EI) concept focused on IoT deployments and some associated research fields, as well as on the obstacles, accessible issues, and potential research opportunities.

The studies for possible IoT uses of power systems in general and in the form of intelligent grids have been reviewed by Motlagh and others [30]. They have also discussed IoT infrastructure assistance, including cloud storage and various frameworks for analysis of data obtained.

Granados [31] and his research team developed a Power-over-Ethernet (PoE) gateway that allowed the Internet of Things (IoT) techniques to support medical sensors and intelligent hospital equipment for both cloud connectivity and energy.

Their proposed design has modified to capture and include health-related data to mitigate the total processing of battery life, overall device power and effectiveness through wireless sensors.

Paul et al. [32] have proposed to use of Fog computing to track and reliably collect and interpret chronically diseased patients. The Internet of Things connects a wide variety of technologies from sensors in addition to computers. Cloud computing networks are used to transmit sensor data to cloud servers and to store it on cloud databases.

Oma et al. [33] have proposed a Tree-Based Fog Computation (TBFC) model for the cloud servers and Fog nodes. The IoT nodes have limited the use of overall electrical consumption and the average electricity consumption of the TBFC. They have observed that TBFC nodes were less than that of the assessment cloud storage model.

4 Proposed Design

This section discusses the proposed design of IoT-based energy harvesting model of the wireless sensor network (EHWSN) in the hospital environment. The development of EHWSN is composed of three subsections. Section 1 describes the proposed designs for wireless sensor networks (WSNs) in the hospital environment. Possible energy consumption modalities in the proposed wireless sensor network (WSN) of the hospital environment (WSNHE) are described in Sect. 2. Section 3 introduces the energy harvesting and management model for the proposed WSN model.

The proposed EHWSN model comprises an energy consumption estimation module, health data processing and decision-making module (HDPDM), sensor data acquisition module, EHIoT module, data analysis module, and remote patient monitoring platform. WSN model composed of a large number of sensors in the hospital environment. Here, medical sensors are utilized to collect health-related data and a human-based energy harvesting process. Figure 2 shows the architecture of the proposed EHWSN system. Table 3 depicts the description of each model proposed in the EHWSN system.

4.1 Design of an IoT/Fog-Based WSN Model for Hospital Environment

This section describes the proposed wireless sensor network (WSN) for the hospital environment (WSNHE), as shown in Fig. 3. The WSNHE composed of sensors, LAN, cloud server, cloud storage, Fog devices, Fog storage, and end-user platforms. The aim is to design an FC-based model for transmitting real-time health-related sensor data to end-users in the hospital environment. Wireless body sensor networks

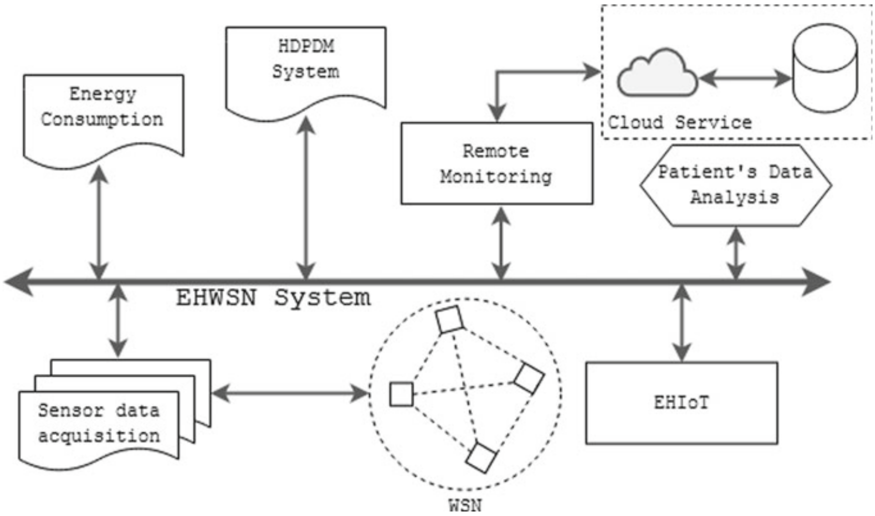


Fig. 2 Architecture of the proposed EHWSN system

Table 3 Actions required to performing tasks in the EHWSN system

Proposed model	Description
Energy consumption	Measure energy consumption of WSN nodes, microcontrollers, and energy usage for network equipments
Sensor data acquisition	Acquired wireless sensor data (medical sensor data such as EEG, EMG, temperature, blood pressure)
HDPDM system	Health data processing (data filtering and classification) and decision-making module
WSN model	Wireless sensor network model in the hospital environment
EHIoT	Ambient-based (wind, solar power) and human-based energy harvesting model (breathing and finger motion)
Patient's data analysis	Analyzed sensor data based on previous data acquired
Remote monitoring model	Monitor and access patents data (finger movement, breathing) in a remote location
Cloud service	Store data on a cloud database and allow end-users to retrieve it whenever necessary

(WBSNs) are used to design the proposed system. We considered the usage of many wireless medical sensors in the hospital environment and here we only discussed wireless medical sensors that have capabilities for generating energy.

The Fog layer contains the health data processing and decision-making module (HDPDM) and the Fog storage, while the server layer holds the cloud storage. Before the processed health data is sent to the cloud for permanent storage, the Fog storage

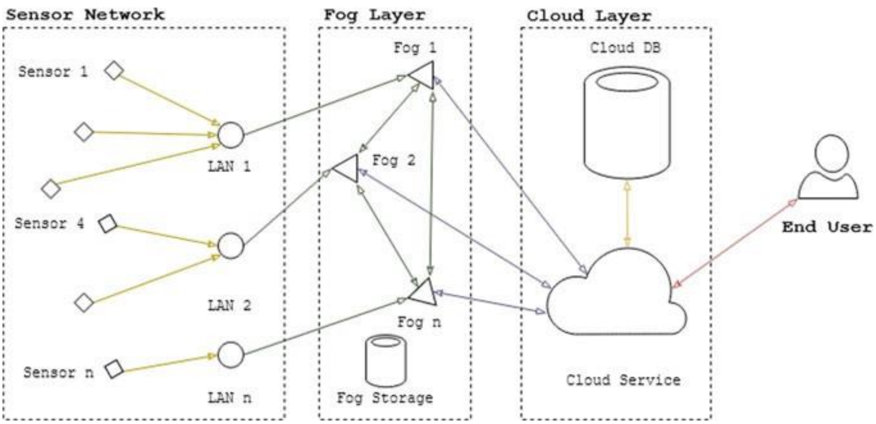


Fig. 3 Proposed WSN-based IoT/Fog architecture of the hospital environment (WSNHE)

module is used as a temporary storage facility. The Fog storage module is available for emergency access to health records.

Three operations are carried out by the HDPDM module. The first is the collection of health-related data from various patients sent via wirelessly linked sensor networks. The second task includes collecting and reviewing the patient's health details and matched the signs of the condition using the derived functions. The final task is to determine what steps to take in reaction to unexpected patient behavior, such as notifying emergency medical service resources, so that they can react effectively to patients in need of assistance. The cloud storage (Cloud DB) module stores the analyzed results of patients permanently as medical history. Similar to the Fog storage module, this module allows functionality for both patients and doctors.

4.2 Design of the Energy Consumption Model

The energy consumption approach of the proposed EHWSN model is briefly discussed in this section. Here, from sensors to servers, Fog nodes are interconnected linearly. For smaller numbers of sensors, the linear model is simple and useful. To reduce the electrical energy consumption of the nodes in the IoT [34, 35], the linear Fog computing (LFC) model is proposed. We proposed a flow-based energy model for WSN since it used heavily shared equipment and dealt with many traffic. We developed an energy use model to figure out how much energy the proposed EHWSN uses in total. Equation (6) shows the derived total energy consumption of the proposed EHWSN model.

$$\mathrm{Total}_E = \sum_{i=1}^{N_M} E(C) + \sum_{i=1}^{N_P} E_{P0} + \sum_{i=1}^{N_T} (E_{T0} + E_{R0}) + \sum_{j=1}^{N} \sum_{k=1}^{N_S} (E_{jk}) + E_K \tag{6}$$

The overall energy ($Total_E$) usage of the proposed design is the combination of energy consumed by the network devices, energy (E_{WSN}) consumed by the wireless sensor network (sensors, transceivers, and microcontrollers), and energy (E_K) consumed by the booster converter, charge converter, and safety circuits. N_M, N_P, N_T, and N_S represent the number of network devices, the number of processors, the number of transceivers, and the number of WSN nodes. Energy storage is a potential alternative to batteries that could enable WSNs to run on their own. When a harvester is used to power a WSN node directly, the instantaneous harvested power $E_H(t)$ is always higher than the instantaneous consumed power $E_C(t)$ Eq. (7):

$$E_H(t) \geq E_C(t) \tag{7}$$

Since the EHWSN system used a large number of WSN nodes to power the entire EHWSN following requirements is fulfilled Eq. (8):

$$\sum_{i=1}^{N} E_H(t)_i \geq \mathrm{Total}_E(t) \tag{8}$$

The next section describes the proposed energy harvesting methodology to harvest efficient energy by fulfilling the requirements mentioned in Eq. (6).

4.3 Design of IoT-Based Energy Harvesting Model (EHIoT)

The EHIoT contains several different energy sources as well as a few transduction processes. In this section, we proposed major energy harvesting methodologies related to the ambient-based and human-based functionalities. Table 4 depicts the EHIoT taxonomy used in this study, composed of two groups (ambient-based and human-based).

The best renewable energy source in the world is solar and wind energy. This type of energy is being researched as a potential source of power for remote wireless monitoring systems. The human body can produce energy in several ways, including finger movements, footfalls, breathing, blood pressure, etc. We proposed solar and

Table 4 Taxonomy of proposed energy harvesting in IoT (EHIoT) model

Energy harvesting Internet of Things (EHIoT)	
Ambient-based	Human-based
Solar energy (PV)	Finger motion (FM)
Wind energy (WE)	Breathing (BR)

wind energy as the ambient-based energy harvesting model, while finger motion and breathing are used as human-based energy harvesting methods.

Finger motions (FM) can be used as a good source of energy (0.76–2.11 mW) [36] for IoT sensors. Additionally, energy can be generated from the breathing process (0.42 W) and this necessitates the use of an external mask. When a band is tied across the chest, breathing is beneficial in the generation of 0.83 W of energy. EHWSN composed of ambient and human-based energy harvesters, boost converter, charge converter, control circuit, battery unit, protection circuit, and WSN nodes. Figure 4 depicts the proposed power supply architecture of the EHWSN system. The control unit comprised of the PID controller and MC unit for control and stabilizes the supply power.

The booster converter connects ambient-based energy harvesters (solar panels and wind turbines) and human-based energy harvesters (breathing, finger motion) to raise the voltage to the required amount to refresh the battery or power the device. If the output voltage (V_{charge}) falls below the amount needed to recharge the battery, the control circuit in the charger circuit disconnects it from the booster converter. The safety circuit is placed on the output of the rechargeable battery device (V_s) to prevent it from being completely discharged. The output voltage from the safety circuit is used to control the WSN nodes. The control circuit can also control the power supply for sensors and actuators. Figure 5 illustrates the schematic of the boost converter for the human-based energy harvester. As seen in Figure, the switches SW1 and SW2 are usually implemented using single nMOS and pMOS transistors.

The PID controller is a basic but accurate and efficient method of system control [37]. The most popular control algorithm is PID, which is the basis for many feedback loop systems. In the EHWSN, a PID controller is proposed to control the energy management of the sensor node.

This design aims to let the sensor node converge to Energy Neutral Operation (ENO). Therefore energy consumed by the nodes should be less or equal to the energy harvested to achieve an effective lifetime. Figure 6 shows the schematic of the PID

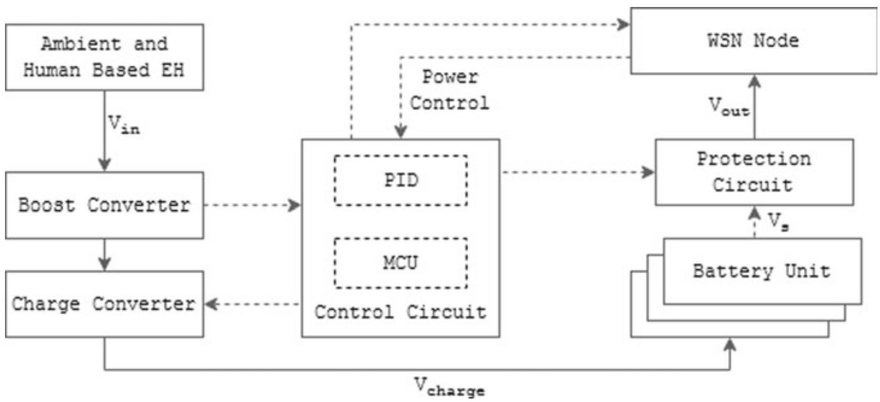


Fig. 4 The architecture of the proposed power supply system

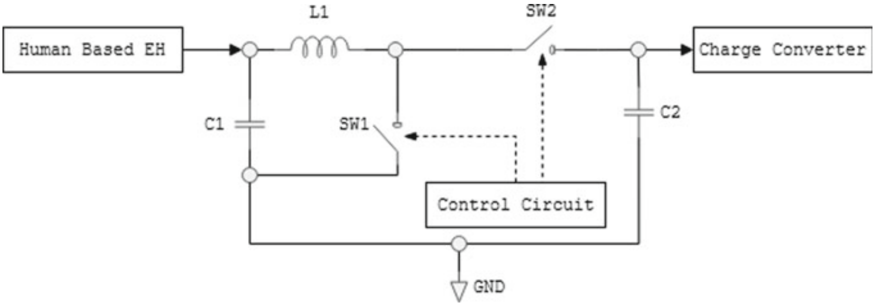


Fig. 5 Schematic of the boost converter (human-based EH)

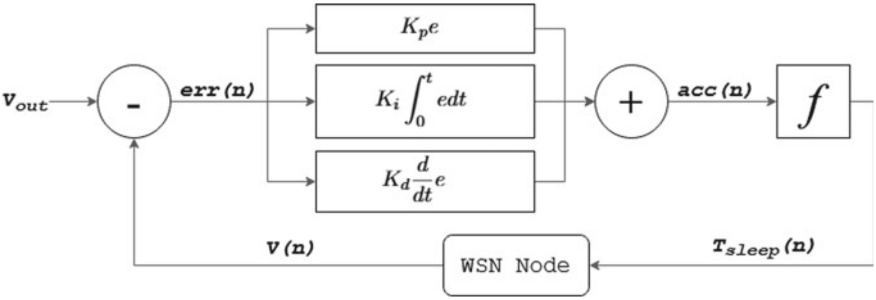


Fig. 6 The schematic of the proposed PID controller used in the EHWSN

controller proposed in EHWSN. The difference between the expected voltage and the stored voltage is the error. The WSN node's sleep time is the controlled output signal. The *f* block is a system-dependent adaptation feature that produces the controlled output signal.

5 Conclusion

Nowadays, health providers, such as modern hospitals use a vast range of wireless sensor networks to gather health-related data from patients [38]. Through IoT technology, modern healthcare services are getting smarter, accurate, and more reliable. Since IoT relies on many devices, sensors, microprocessors, etc., to deliver a service, it requires a large amount of energy. To meet the energy demands in the healthcare sector, energy harvesting (EH) technologies used by IoT devices and sensors would improve EH processes. In this chapter, we proposed a design model for the integration of the wireless sensor network (WSN) for the hospital environment (WSNHE) by enabling IoT-based energy harvesting (EHIoT) methodologies. Ambient-based and human-based energy harvesting techniques were considered to generate energy for

WSN. The boost converter was designed to step up the supply voltage to the charging circuit. PID controller was designed to control and stabilize the power supply to the WSN in the proposed system.

References

1. Zanella A, Bui N, Castellani A, Vangelista L, Zorzi M (2014) Internet of things for smart cities. IEEE Internet Things J 1:22–32
2. Bui N, Castellani AP, Casari P, Zorzi M (2012) The internet of energy: a web-enabled smart grid system. IEEE Netw 26:39–45
3. Minoli D, Sohraby K, Occhiogrosso B (2017) IoT considerations, requirements, and architectures for smart buildings—energy optimization and next generation building management systems. IEEE Internet Things J 4:269–283
4. Razzaque MA, Milojevic-Jevric M, Palade A, Clarke S (2016) Middleware for internet of things: a survey. IEEE 3:70–95
5. Arcadius Tokognon C, Gao B, Tian GY, Yan Y (2017) Structural health monitoring framework based on internet of things: a survey. IEEE Internet Things J. 4:619–635
6. Zhao K, Ge L (2013) A survey on the internet of things security. In: 9th international conference on computational intelligence and security (CIS), IEEE, pp 663–667
7. Singh R, Gahlot A, Mittal M (2019) IoT based intelligent robot for various disasters monitoring and prevention with visual data manipulating. Int J Tomogr Simul 32(1):90–99
8. Sheng Z, Yang S, Yu Y, Vasilakos AV, McCann JA, Leung KK (2013) A survey on the IETF protocol suite for the internet of things: standards, challenges, and opportunities. IEEE Wirel Commun 20(6):91–98
9. Vasseur JP, Dunkels A (2008) Ip for smart objects. White Paper, 1, pp 1–6
10. Duolikun D, Kataoka H, Enokido T, Takizawa M (2018) Simple algorithms for selecting an energy-efficient server in a cluster of servers. Int J Commun Netw Distrib Syst 21:1–25
11. Singh R, Gehlot A, Mittal M, Samkaria R, Choudhury S (2017) Application of icloud and wireless sensor network in environmental parameter analysis. Int J Sens Wirel Commun Control 7(3):170–177
12. Saleem MA, Zhou S, Sharif A, Saba T, Zia MA, Javed A et al (2019) Expansion of cluster head stability using fuzzy in cognitive radio CR-VANET. IEEE Access 7:173185–173195
13. Enokido T, Aikebaier A, Takizawa M (2010) Process allocation algorithms for saving power consumption in peer-to-peer systems. IEEE Trans Indus Electron 58:2097–2105
14. Bonomi F, Milito R, Zhu J, Addepalli S (2012) Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on mobile cloud computing, pp 13–16
15. Jaouhari SE, Palacios-Garcia EJ, Anvari-Moghaddam A, Bouabdallah A (2019) Integrated management of energy, wellbeing and health in the next generation of smart homes. Sensors 19:481
16. Gia TN, Jiang M, Rahmani AM, Westerlund T, Liljeberg P, Tenhunen H (2015) 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
17. de Morais Barroca Filho I, de Aquino Junior GS (2017) IoT-based healthcare applications: a review. In International conference on computational science and its applications. Springer, Cham, pp 47–62
18. da Costa CA, Pasluosta CF, Eskofier B, da Silva DB, da Rosa Righi R (2018) Internet of health things: toward intelligent vital signs monitoring in hospital wards. Artif Intell Med 89:61–69

19. Mouradian C, Naboulsi D, Yangui S, Glitho RH, Morrow MJ, Polakos PA (2017) A comprehensive survey on fog computing: state-of-the-art and research challenges. IEEE Commun Surv Tutorials 20(1):416–464
20. Elahi H, Munir K, Eugeni M, Atek S, Gaudenzi P (2020) Energy harvesting towards self-powered IoT devices. Energies 13(21):5528
21. Alam MGR, Munir MS, Uddin MZ, Alam MS, Dang TN, Hong CS (2019) Edge-of-things computing framework for cost-effective provisioning of healthcare data. J Parallel Distrib Comput 123:54–60
22. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) Wireless sensor networks: a survey. Comput Netw 38(4):393–422
23. Kahn JM, Katz RH, Pister KS (1999) Next century challenges: mobile networking for "smart dust". In: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, pp 271–278
24. Herath HMKKMB, Ariyathunge SVASH, Priyankara HDNS (2020) Development of a data acquisition and monitoring system based on MODBUS RTU communication protocol. Development 5
25. Jalali F, Hinton K, Ayre R, Alpcan T, Tucker RS (2016) Fog computing may help to save energy in cloud computing. IEEE J Sel Areas Commun 34(5):1728–1739
26. Vishwanath A, Jalali F, Ayre R, Alpcan T, Hinton K, Tucker RS (2013) Energy consumption of interactive cloud-based document processing applications. In: 2013 IEEE international conference on communications (ICC). IEEE, pp 4212–4216
27. Jalali F, Gray C, Vishwanath A, Ayre R, Alpcan T, Hinton K, Tucker RS (2014) Energy consumption of photo sharing in online social networks. In: 2014 14th IEEE/ACM international symposium on cluster, cloud and grid computing. IEEE, pp 604–611
28. Hashmi SWA, Alvi BA, Rehan M, Kamal MS, Zaheen MY (2014) Energy efficient vital signs monitoring system (VSMS) using wireless sensor networks. Wireless Pers Commun 76(3):489–501
29. Kabalci Y, Kabalci E, Padmanaban S, Holm-Nielsen JB, Blaabjerg F (2019) Internet of things applications as energy internet in smart grids and smart environments. Electronics 8:972
30. Hossein Motlagh N, Mohammadrezaei M, Hunt J, Zakeri B (2020) Internet of things (IoT) and the energy sector. Energies 13(2):494
31. Granados J, Rahmani AM, Nikander P, Liljeberg P, Tenhunen H (2014) Towards energy-efficient healthcare: an internet-of-things architecture using intelligent gateways. In: 2014 4th international conference on wireless mobile communication and healthcare-transforming healthcare through innovations in mobile and wireless technologies (MOBIHEALTH), IEEE, pp 279–282
32. Paul A, Pinjari H, Hong WH, Seo HC, Rho S (2018) Fog computing-based IoT for health monitoring system. J Sens
33. Oma R, Nakamura S, Duolikun D, Enokido T, Takizawa M (2018) An energy-efficient model for fog computing in the internet of things (IoT). Internet Things 1:14–26
34. Enokido T, Aikebaier A, Takizawa M (2010) A model for reducing power consumption in peer-to-peer systems. IEEE Syst J 4:221–229
35. Abdussami AA, Farooqui MF (2020) A systematic literature review on fog computing. Int J Adv Sci Technol 29:12750–12769
36. Castagnetti A, Pegatoquet A, Belleudy C, Auguin M (2012) A framework for modeling and simulating energy harvesting WSN nodes with efficient power management policies. EURASIP J Embed Syst 1:1–20
37. Persson E (2019) Energy harvesting in wireless sensor networks, Diva-Portal, 1–70
38. Herath HMKKMB, Karunasena GMKB, Herath HMWT (2021) Development of an IoT based systems to mitigate the impact of COVID-19 pandemic in smart cities. In: Machine intelligence and data analytics for sustainable future smart cities, Springer, pp 287–309

Design and Development of Efficient Secure Routing Mechanism for Wireless Sensor Network

N. L. Taranath, H. R. Roopashree, A. C. Yogeesh, L. M. Darshan, and C. K. Subbaraya

Abstract The prime research goal of the study is to address the security problems in WSN communication in order to safeguard information exchange over various potential critical applications integrated with different radio-frequency (RF) channels. It has also evaluated the background of security limitations in WSN by investigating significant scientific literatures and its impact on other performance aspects from energy viewpoint. The design limitations associated with the existing secure routing approaches shows that most of the working principles of routing only limited toward privacy protection of data and the importance of energy consumption minimization are to some extent ignored and overlooked. Thereby, the research goal defined in this study is to minimize the security loopholes in WSN with higher degree of intrusion detection and prevention. Thereby, the study formulates a novel tree-based strategy which is exclusively meant for secure and energy-efficient hierarchical routing in WSN. The system modeling of the formulated approach consists of three different solution spaces such as (i) a framework for energy-efficient secure routing (FEESR), which incorporates a novel tree-based approach is introduced in this study, followed by two more approaches such as (ii) delay sensitive protocol for intelligent routing (DSP-IR) and an optimized and roust sandboxing approach (ORSA). All the solution approaches are designed in a way where these all jointly addresses the security problems of WSN without compromising the energy aspects. The study introduces a novel optimization policy to balance the trade-off between energy and security aspects. The experimental simulation of the proposed analytical modelings is carried out in a numerical computing platform with respect to mathematical computation

N. L. Taranath (✉) · H. R. Roopashree · L. M. Darshan
Department of CS & E, Adichunchanagiri Institute of Technology, Chikkamagluru, Karnataka, India

A. C. Yogeesh
Department of CS & E, GSSSITEW, Mysuru, Karnataka, India

L. M. Darshan
Department of CS & E, Government Engineering College, Kushalnagar, Karnataka, India

C. K. Subbaraya
Adichunchanagiri University, B.G Nagar, Javarana Hally, Karnataka, India

R. Tiwari et al. (eds.), *Energy Conservation Solutions for Fog-Edge Computing Paradigms*, Lecture Notes on Data Engineering and Communications Technologies 74,
https://doi.org/10.1007/978-981-16-3448-2_12

principle. The performance validation of the models FEESR, DSP-IR, and ORSA are done with respect to different significant parameters where the comparison has been done with the most significant energy-efficient and secure routing baselines. The Simulation outcome obtained for all the analytical models shows that the study outperforms the conventional baselines from both energy and security viewpoint.

1 Background

The motivational factors with a strong vision behind establishing sensor-driven communication and networking has been derived many years back to avoid catastrophic infrastructure failures, preserve significant natural resources, etc. It has a broad range of applications including context aware technologies and smart-home technologies [1, 2].

(a) Sensor Node Overview

The advancement in various integrated systems such as very large-scale integration (VLSI), micro-electro-mechanical systems (MEMS) along with wireless communications has led to miniaturization of computing and low-power devices. A sensor node is a tiny and low-power computational device that intends to perform sensing in a distributed widespread region with the ease of low-cost implementation. The following Fig. 1 displays an overview of a MicaZ sensor node with different computing and communication units [3, 4].

Sensor nodes also deal with data acquisition and actuation where the actuators directly allow them to control the physical world data. However, a set of sensor nodes working collaboratively in a large-scale area to accomplish a particular defined task consist of a typical WSN. The following segment thereby introduces fundamental aspects of a typical WSN which makes it different from traditional networks [5].

(b) WSN Overview

WSNs are mainly adopted for environment monitoring purpose. The WSN sensor node detects event from the placement region. Afterward, it collects data from the surroundings, and processes the collected data and further transmits it to the sink or base station. Today the application of WSN is widespread in many areas like wildlife, ocean, manufacturing, earthquake, national border security monitoring systems. In

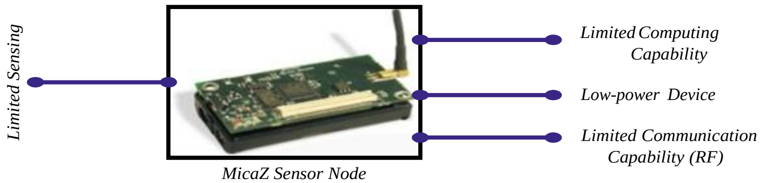


Fig. 1 A typical MicaZ sensor node

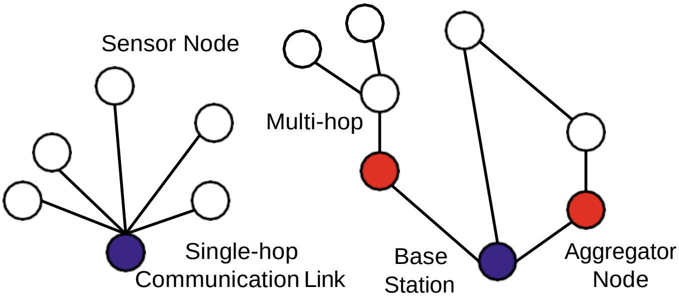


Fig. 2 A tree structure-based topology oriented WSN

future, applicability of the WSN includes the monitoring of vehicle traffic system, record pollution monitoring, fire monitoring for the forest, measurement and monitoring the human and animals heart rates, etc. The major advantages of this WSN are on network application. The following Fig. 2 shows single-hop Vs multi-hop communication associated with a WSN. It also shows how a tree-based structure can be used with number of vertices and edges to deploy a sensor network topology.

2 High-Level Techniques (HIT)

Advanced security mechanisms for protecting sensor networks incorporate security group management, IDs (intrusion detection system), and secure data fusion and aggregation. This technique is used to protecting the sensor networks, including data aggregation, group management, and intrusion detection.

- **Secure Data Aggregation**: The sensed data needs to be aggregated before being sent to the BS to avoid congestion and overhead issues at the BS. The aggregator node collects data from multiple locations and sends the aggregate data (i.e., non-redundant) to the BS. Depending on the WSN architecture, aggregation can occur in many places in the network. There must be a protocol that protects each aggregation location.
- **Securing group management**: Each node in WSN has limited communication ability. A standout among the most difficult issues in sensor systems is strength against hub catch assaults. In many applications, sensor hubs are liable to be set in areas effortlessly open to assailants. Such introduction raises the likelihood that an aggressor may catch SNs, separate cryptographic privileged insights, adjust their programming. However, it is costly since current innovation does not give an abnormal state of security. Algorithmic answers for the issue of hub catch are ideal.

- **Intrusion detection**: WSNs are susceptible to very intrusion. WSNs require a solution which is less expensive and distributed regarding memory energy requirements and communication. Adoption of the security group mechanism may be a promising approach to distributed intrusion detection [6].

3 Problem Description

The security mechanism for WSN should ensure the security of sensitive network channels and data transmitted under various security risks. The following points provide a brief overview of various types of attacks, which are as follows: Attacks on the physical layer—The attacks launched on this layer lead to serious impact, which results in the blocking of entire services and network resource usage. Attack on the link layer—In this, the intention of the attacker is to eliminate all the resources used by the network component and services. Attack on Network Layer: There are too many attacks on the network layer. There are various types of attacks and types of attacks (Sinkhole, Blackhole, Wormhole, etc.), which take different actions to harass all network services and to steel important information. Attack on transport layer—In this, network services are unavailable for some time and are available for some time. This is an unexpected type of attack, the analysis of these changes is quite difficult due to its variable nature. In this, unnecessary traffic is released between sensor nodes and confuses the entire operation of communication and data transmission processes. Many such attacks are still open to explore more and its effective solution remains a challenging factor in the WSN research community, especially when it is linked to odd WSN. Another serious factor related to the WSN research sector is that there are very few system that considers BS Security, because BSS is a main junction for data gathering in the WSN, if there is no presence of strong security implementation, then it becomes the central point for the attackers, and they can perform various types of unwanted tasks. The study also realized that the existing security solutions provided by the hybrid version of LEECH address only a specific form of attacks. Hence, the problem statement can be illustrated as *Developing an efficient framework is a very challenging task that can maintain an improved performance of security without negatively affecting the performance of the communication.* The upcoming section highlights about proposed system and its contribution [7].

4 Study Objectives

The key purpose of the chapter is to design and develop a tree-based scheme for modeling innovative framework of robust and energy-efficient hierarchical routing protocols in WSN. In order to achieve the above demonstrate research goals, the following objectives are considered:

- Survey
- To establish tree-based routing scheme based on concept of probability theorem to acquire highest energy efficiency and robust authentication procedure in WSNs
- To design a delay-based routing technique using tree structure that can further optimize the security and energy efficiency
- To design and develop a simple and lightweight optimization protocol that can enhance the performance of the energy efficiency and safe routing.

5 Literature Review

5.1 Literature Survey Based on Hierarchical Routing Protocols

The very distinguished and unique characteristics of WSN that include its architectural constitutes of operational units of nodes and sink operating at low power for sensor data gathering, processing then communicating require a specific design approach for its routing protocol to meet the energy balance goals. The popularity of the WSN is growing rapidly so that large-scale WSNs are required for many critical applications and gaining scalability into a resource constraints network is quite a challenging task. Many kinds of literature reveal that an approach of hierarchical routing provides the promising result to balance the energy and scalability trade-off [8–12].

The traditional benchmark hierarchical routing protocols are from LEACH or its ingredients [13]. In all these descendants of Leach adopts a processing clustering the entire deployment location and election of a node where the data collection, gathering, fusion or aggregation takes place as per the define objectives of the applications. Few important studies include work on variants of LEACH [14–43].

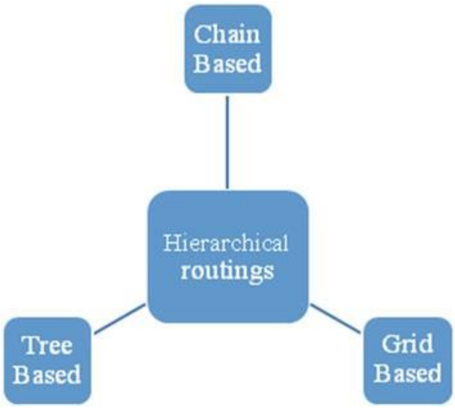
Further, by analysis, the pros and cons of the LEACH-based protocols yet another few important types of hierarchical routings are being evolved that typically includes (1) chain, (2) grid and (3) tree-based routing design as shown in Fig. 3, apart from clustering and area-based routing.

The research study focuses on the tree-based routing protocol as one of its objectives to meet the research goal. Section 5.1.1 elaborates the tree-based routing.

5.1.1 Tree-Based Routing

Four popular tree-based routing protocols are referred for a study that includes (1) EADAT by Ding et al. [44], (2) BATR by Kim et al. [45], (3) PEDAP by Tan et al. [46], (4) ETR by Qiu et al. [47].

Fig. 3 Types of hierarchical routings



Energy-Aware Distributed Aggregation Tree (EADAT) [44]

EADAT is an energy-efficient distributed aggregation tree protocol which is controlled in a distributed manner with no mobility support. EADAT moderate in energy efficiency with less implementation cost as well provide moderate energy efficiency and node balancing with large delivery delay and low scalability and algorithm complexity. The typical process of EADAT includes a mechanism of scheduling where every node exploits their neighbor node to propagate control messages consisting scheduling information. Figure 4 shows the implementation and methodology adopted for the EADAT.

To perform energy-efficient data-centric routing first has to create the network modeling. The approach is concentrated on anomaly event monitoring on communication in WSN. First has to assume that there are thousand or more than sensor has to deploy in the interested area. The gateway is placed at the targeted area which connected to the Internet (outside the distributed system) through some reachable sensors. The wireless sensors or microsensor are considering an event source and gateway is taken as an event sink. The microsensor contains four segments: sensing, battery source, digital processing, and radio transmission/receiver. The radio transceiver is required more power during process; however, it can overcome by sensors.

The microsensor network in data-centric routing involves two kinds of traffic that can observe user to network communication for queries and information from the sensor to the consumer. So here the sensor has to acts as a plain sensor to explore the environmental requirements but these communications happen by one sensor is only large and it can block the whole network medium. Hence the prior exhibited data in the medium may have a chance to get the loss. In this case all unwanted and important data everything is jumbled from the various nearest sensors then aggregated. So data-centric routing improves the performance. The data is tracked by a multicast tree in which sink considers as root.

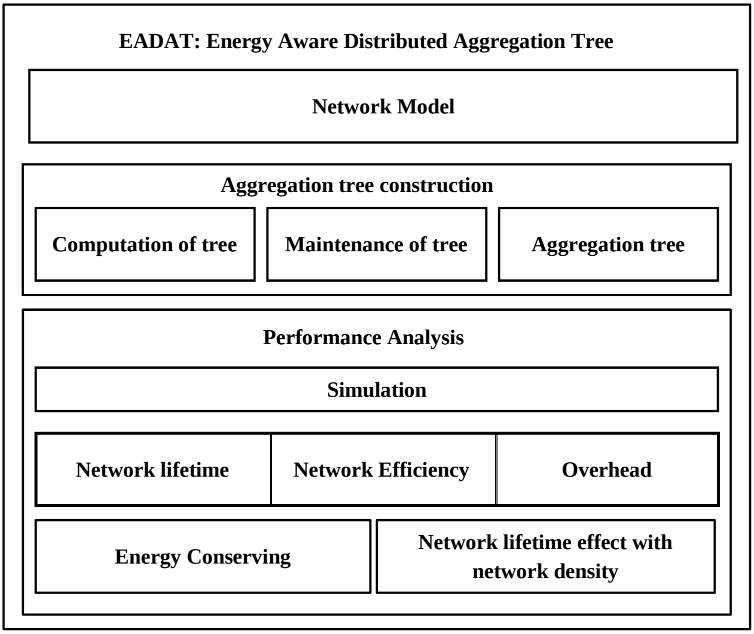


Fig. 4 Architectural building blocks of the methodology adopted for EADAT

The aggregation tree computation is an important task for data-centric routing. The aggregation tree construction initially assumes that each sensor has equal transmission distance, and radio trans-receiver is on when the network is primarily arranged. The network of each sensor does not compulsorily have a unique ID and out of the network or neighbor sensor node has different or unique sensor IDs. The communication control message has five segments such as sensor ID, status in the tree likewise leaf, non-leaf node, and undefined state, parent node in the tree, residual power, and path length. The residual power and time are inversely proposal each other, i.e., the residual power is high then the timer is small ($T \propto 1$/residual power).

In this, the sensor has a higher value of residual power there is a chance to be non-leaf tree node. The neighboring sensors communicate with each other in a circulated manner and the winner will broadcast the control packet based on the channel root. If every sensor selects one more parents node (two) based on residual power it goes to data aggregation tree without changing the initial protocol. Also, it performs multiple message transmission when a message is short using little more power. The aggregation tree construction period the EADAT has showcased their performance in each sensor node to broadcast the information.

The maintenance of the aggregation tree is also an important thing by the heuristic. If the state is in danger using parent node and other significant factors, the message is sent the hello message through the sensor to sink node and invite to join the tree. By applying aggregation tree to the dangerous sink node to address root for perfect communication. But without knowing the interesting point for communication will take more time, B.W. and cost, i.e., it has to send data globally. Sometimes the queried sensor is not active then the nearby sensor takes care of that data until the targets sensor becomes active. The data-centric aggregation tree is also performed as first all sensor takes the information and aggregates them and then sends to the interested sink node.

Finally, the demonstration of methodology has been done by simulation through NS2 simulator. The performance of the EADAT is calculated where the network lifetime is improvised, the energy consumption is low, network efficiency is high, network lifetimes is parallel enlarged by enhancing the network density, and the decrement of overhead. Hence it concludes that the exhibited method is stable.

Balance Aggregation Tree Routing [BATR] [45]

Balance Aggregation Tree Routing [BATR] aims for optimal design of the aggregation process using a tree structure for enhancing the network lifetime by reduced use of energy at the local level. It ensures consistent distribution of traffic demand irrespective of the sink location. The design aspects of BATR also do not support mobility and control in a circulated manner with a facility of data aggregation, whereas the energy efficiency, scalability, and implementation cost lower with moderate algorithm complexity but very poor load balancing capacity (Fig. 5).

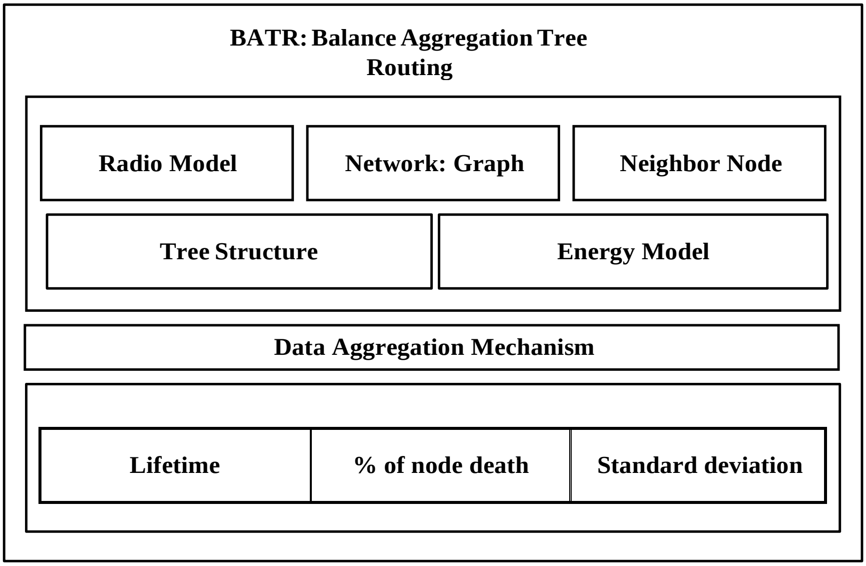


Fig. 5 Architectural building blocks of the methodology adopted for BATR

The data aggregation is processes of combining the data's from multiple sources and reducing the energy usage in terms of decreasing the messages in the network but little energy used in the intermediate node. First model the graph using spanning tree algorithm then define the edge using size and packets. Where the energy cost is depending on two contents for transmitting and receiving in radio electronics and the other thing is depending on the transmission distance, minimize the number of nodes while transmitting the distance is minimizing for a particular node.

The routing protocol base data packet transmitting from the source node to base station maximizes the network lifetime of the medium is concentrated here. For improving the lifetime of network the power loss should be less in the communication among the sensor nodes. The aggregation tree base balance trees with fewer edges are shown as a similar number of child nodes to enhance the network life period. The child node will appear in a neighborhood. This protocol design shows the right path using a balanced tree where almost the same number of a child node is using in the flow. The design process of this first assumes then the sensor in the system and which are randomly roaming in the surrounding. The base station of the network knows all the information among the node communication through GPS or manually communication. In this, the sensor nodes sense surrounding environments to send the data to a base station. It assumes that the sensor node and sink node have the same transmission range and all nodes generate an equal amount of data without any mobility in the network. The minimum spanning tree algorithm is used for routing information computation.

Firstly, the construction of a path has started through the base station as root then in every repetition the minimum weighted edge is selecting. The used protocol is to balance the childhood nodes which are contingent on the density of the medium it helps to enhance the network life period. Based on the set of rule, when the neighbor node is found over the limit of search, the node is labeled as a leaf node and search limit is considered as 15, it will circulate up to every node are added to the tree. Hence after some iteration the base station re-compute the routing information except the dead node and child node resetting. Once the computation in each step is done the base station sends to a node to send the information as a parent node in the tree to transmit to BS. The cost function and energy consumption is achieved very less in this system after every computation.

The outcome of the present method is performed in terms of a lifetime for a 100X100 network where BS is placed at [39], [39, 200]. Next, the lifetime vs numbers of child nodes is graphically represented when the base station is far from the field and center the field. The standard deviation of the node is also calculated, and it shows that the method is efficient.

Power Efficient Data Gathering and Aggregation Protocol [PEDAP] [28]

- **PEDAP** uses a Minimum Spanning Tree to realize the routing protocol where the energy minimization is achieved by equal distribution of load throughout the network. It is a centralized controlled architecture with data aggregation capacity

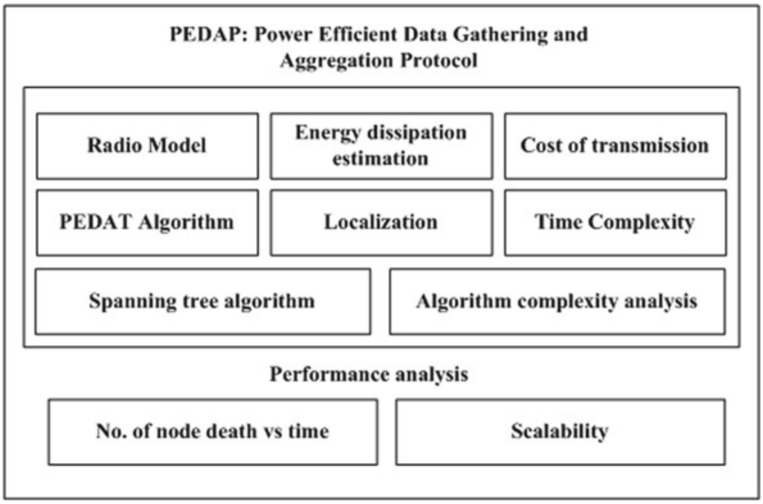


Fig. 6 Architectural building blocks of the methodology adopted for PEDAP

along with mobility support. The energy efficiency, delivery delay, and algorithm complexity are found to moderate, whereas scalability is not so promising, but the load balancing is better (Fig. 6).

Initially, the PEDAP protocol considers some assumption that the node location is already known based on the base station. The centralized algorithm is utilized to compute routing information because the system is resource limited and some of the elements are powerful which are necessary to provide the computation load to the system (powerful elements) where minimum spanning tree is utilized to determine the routing information. The algorithm working is illustrated in above BATR. The spanning tree computes over the network graph with a BS which achieves a minimum energy-consuming system. According to this cost model, the energy consumption or loss of every node in each iteration can be calculated before only. Later some iteration the base station also recalculates the routing information except the formed dead node. Once the computation of each is done then the base station sends wanted information to each node for that parent node. Hence it concludes that the cost of the presented system including routing information is equal to the summation of the receiver circuitry cost for every single node. Also, it says that the periodically published system setup cost is small than LEACH and PEGGASIS by the comparison results.

The present work contains two algorithms with the same protocol and it shows the changes in cost. Also, it makes changes in base station and there is no change in the sensor node. So the system with the presented algorithm is more applicable when the various application and lifetime requirements with the same sensor network in time to time. The simulation is done through C and ns (network simulator) that results that the algorithm is more effective in real-time environments. The performance of the present algorithms is calculated by comparing the results with the direct transmission, PEDAP, LEACH, PEGASIS, and PEDAP-PA. The number of node death vs. time taken is calculated for each algorithm with the graphical representation (base station distant far from the fields and center with different sizes 50 × 50, 100 × 100).

Enhanced Tree Routing (ETR) [47]

ETR exploits a linking pattern among the child node and the parent node for finding the shortest path with a decision-making mechanism of structuring the object to ensure optimal use of storage and computational resource. It is a tree base distributes control mechanism with moderate energy efficiency, scalability, and delay, whereas the load balancing capability is not adequate enough. It does not support mobility, but the lower complex algorithm performs data aggregation at the minimum cost.

The ETR protocol is used to balance between performances and cost by enhancing the hop counts of tree routing with some minor complexity. The structured address assignment scheme is utilized in the process of ETR. The packet travels to reach a destination where the parent–child node will choose the next hop. One of the main parameters connected to each node in a medium is the network complexity, in which a small amount of hops is used to send the transmitted node to the root node using the parent–child link.

The network depth 0 in root node and network depth is not zero in the non-root node. If the communication process gets easy or it should be peer-to-peer among the nodes, then every node must have a unique ID or it has a network address. The ad-hoc nature of the network topology an address assignment scheme has to be in place to allocate addresses to the node to connect the network.

The performance evaluation of the ETR is tested against the TR protocol. The structured deployment and random deployment is considered in the test to show the performance or character of ETR. The event-driven simulator implemented in MATLAB (Fig. 7).

It is being observed after analysis of these four methods that it could not be made scalable topology because it uses a large number of communication hops which introduces delay.

5.2 Existing Research Work on Delay Concept in Routing for WSN

The wireless sensor network is expected efficient routing, reliable and real-time routing to transfer data from source to destination node within some deadlines (without delay). The packet delivery gets some delay in between and which dynamic and uncertainties are because of factor such as queuing and spatiotemporal dynamics. Hence, these uncertainties and dynamics in path delay present many challenges to

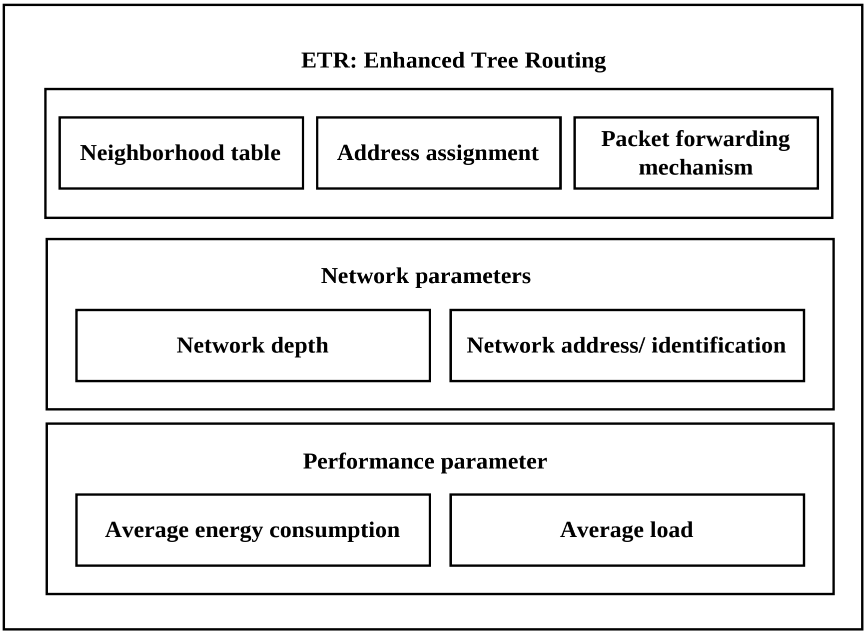


Fig. 7 Architectural building blocks of the methodology adopted for ETR

routing in real time. So many prior existing works reveal that delay reduction base routing approaches introduce efficient results to balance the reliability, scalability, and energy efficiency [48]. Most of the existed wireless sensor routing protocol with delay aware approaches [49–53] that deliberate average delay instead of delay quantiles.

- The multi-time scale estimation approach [54] to efficiently exhibit the mean and variance of packet transmission time with adapting fast queuing process. With the help of a multi-timescale approach, it is addressed which assist to real-time information delivery in the presence of uncertainty and dynamics. The Directed Acyclic Graph is also played the main role in this to support the routing protocol. The method performance is calculated using the sensor network bed NetEye and TelosB. From the results, it shows that time scale adaption protocol exhibited better performance than other protocols by reducing the transmission cost and deadline success ratio enhancement.
- The flow of the proposed method is represented as a system model, cognitive networking with routing protocol and performance analysis. In the system design the network address mechanism, radio implementation, and link model are elaborated using various criteria. The link model performed involved three factors for successful packet transmission among two nodes. Such factors are channel availability, channel access priority, and packet reception ratio. Next, cognitive networking.

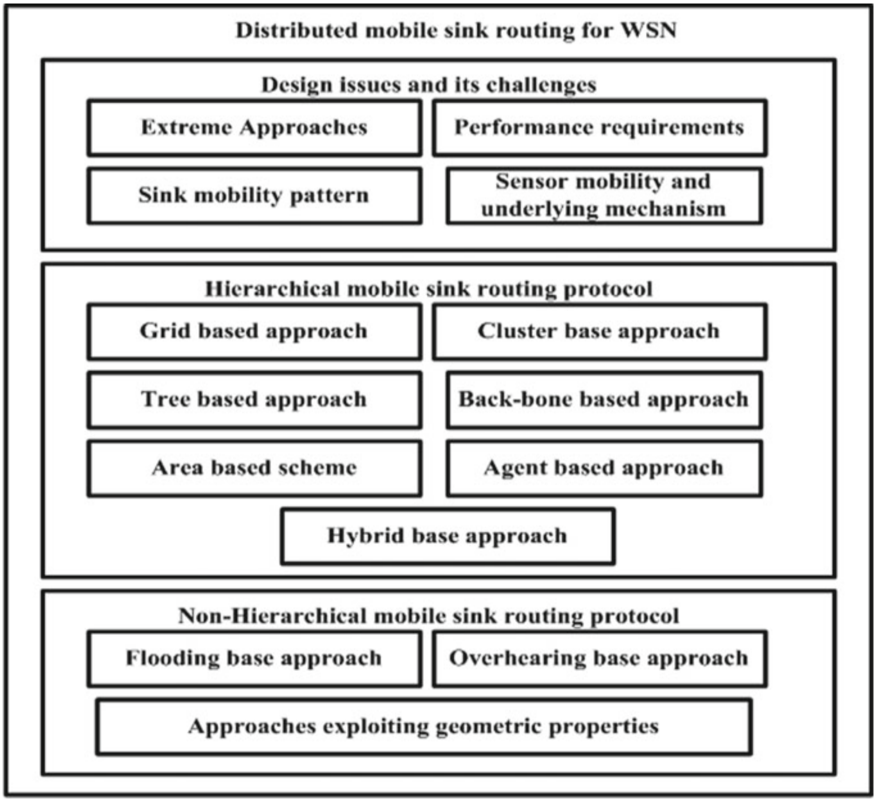


Fig. 8 Architectural building blocks of distributed mobile sink routing protocol survey work

- The WSN is playing a plethora of application in different fields. To provide these the network must be efficient so the cognitive radio capability is incorporated to sensor network which enhances the network performance [55]. Cognitive network routing protocol performance is evaluated through discrete event simulator and compared with other two traditional protocols. It shows that the addressed protocol is excellent concerning throughput, power consumption, and packet delivery delay (Fig. 8).

6 Energy-Efficient and Security-Aware Routing Protocols

6.1 FEESR Design Methodology

The study aims to overcome the limitations associated with the improved LEACH-based routing solutions, thereby the motivational factors this study lies in the fact which shows the recent improvement in the hardware-based public key cryptography approaches. It can be also observed that majority of microcontroller-based cryptosystems attain faster response time, irrespective of the size of public key. The design goal of FEESR is to strengthen the data security in WSN while not compromising the Quality of Service (QoS) aspects. The QoS aspect is the key attribute to measure the communication performance of a WSN. It also aims to make the framework resilient against different form of security attacks such as message corruption and tampering attacks.

As the security mechanism of sensor node mostly constructed on public key cryptography, the sensor nodes have to share their key to perform their respective operation. The process of key generation and sharing become an initial point of attack. It has also been observed that most of studies have not focused to utilization of base stations because it is only used as a junction of data gathering. But the base stations contribution is quite more that it is also performed as trusted authority. Also, Secure routing toward BS has not much-received attention they all only concentrated to build secure routing toward CH and the sensors. Thus, the design and modeling of FEESR state that if there is unlimited energy in a base station, then it can be utilized to perform a key role in security, although it cannot help a lot in the data collection process. Therefore, this chapter addresses an innovative approach in which a secure routing plan for heterogeneous WSN is discussed. The proposed plan also supposed to deliver stable balance between efficient energy utilization, quality ware performance and secure routing.

- Using Base Station to authenticate the cluster key and revocation list generation.
- The study considers static phase and mobility phase of nodes with innovative mobility framework constructed for Heterogeneous WSN.
- FEESR also performs a job of surveillance over node mobility for security reason and also performs fine-tune operation on node key generation process according to report of revocation list and node identifier.
- The study utilizes lightweight cryptography concept to make sure that nodes do not deplete extra energy to perform and process security protocols. Hence, consistency and stability are balanced among the communication quality and security measures.

The proposed study is constructed and formulated using analytical concept where concentration is fully placed on safe routing process. The study considers a wide area WSN with both static and mobile nodes.

The framework is displayed in Fig. 9. The model constructed on simulation strategy by taking input variable as Member Node (MN), Cluster Head (CH) and Base Station. The initial phase of the design and development of FEESR considers deploying different types of node to mimic the hypothetical heterogeneous WSN operations (Fig. 10).

In Fig. 10, it is shown that the FEESR node deployment phase considers two different types of sensor node where in the first type, sensor node with higher computing configuration is considered, where in second type, sensor node with lower computing configuration is considered.

The model mainly involves dynamic key management schemes where the development of secret keys is mainly emphasized. A cluster key is implemented to encrypt

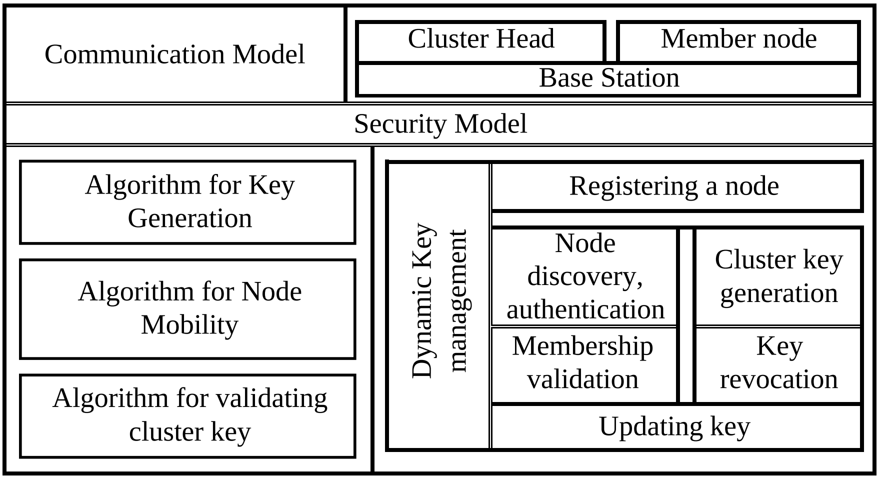


Fig. 9 Framework of FEESR

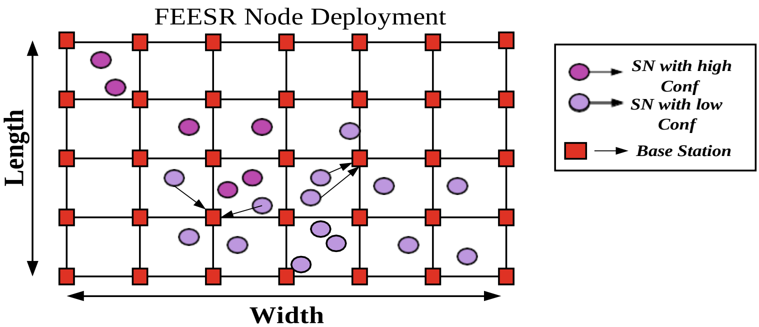


Fig. 10 FEESR Sensor node deployment scenario

the beacon generated and shared by the cluster. Only CH is responsible to perform updating of the cluster key. Now the next step is about to obtain all the registered nodes by the trusted authorities that shares split keys. Once all the keys have been developed. The node discovering and node validation is done by CH, that also responsible for creating cluster keys-based lightweight hash algorithm. Once the neighbor nodes are identified, the CH calculates the key and sends it back to the BS. The base station authenticates the key with its list of key details. The complete encrypted key is updated so that it can be ensured that there is no stale key utilizing the framework. All of the operations are executed through three important algorithms as demonstrated in figure shown in Fig. 21, and it will further discuss in the next section. The key feature of proposed FEESR is that one can easily recognize new nodes joining and old nodes leaving CH network. The study adopts a lightweight cryptographic mechanism for constructing the robust routing in Heterogeneous WSN. The next section illustrates implementation of algorithm in FEESR.

6.2 FEESR Algorithm Design and Implementation

This section addresses the procedural steps to build the stable strong, energy saving mechanism, and lightweight secure routing in Heterogeneous WSN.

Algorithm for Key Generation

Input: K, P_{key}, M_{pk}, P, θ, n, MN

Output: key

Start

1. init K, n
2. $P_{key} \rightarrow \alpha$ where $\alpha \sim M_{pk} * p$
3. $n_i \rightarrow$ select(PRNG(m)), where M=1-1000
4. Compute public & private key

    $Pub_{key} \leftarrow [\, P_{key}, (1+999*PRNG(MN))]^T$

    $Pri_{key} \leftarrow [\text{split key}]^T$

5. Compute key

key→hash(PRNG, PRNG(MN))

End

The above algorithm determines public key based on the multiplication of unique master private keys on *P* (point generator finite area) by using different number of system parameter *K* (step-2). Therefore, it can be stated that the private master key can be also be utilized to assess the public key. The next step is to perform public key generation for member nodes where it will select pseudo-number generator PRNG from node 1–1000 (step-3). The base station then executes an integrated calculation of the public and private keys, as demonstrated in the step-4. However, one crucial factor is that the BS does not calculate the complete calculation of both, but it performs only the partial calculation. It is main one of the unique contributions that have two advantages, i.e., (i) Safe from content modification and guessing attacks, (ii) It reduces the key size based on each request. The following Fig. 23 demonstrates computational steps involved in estimating the numerical design for the key generation (Fig. 11).

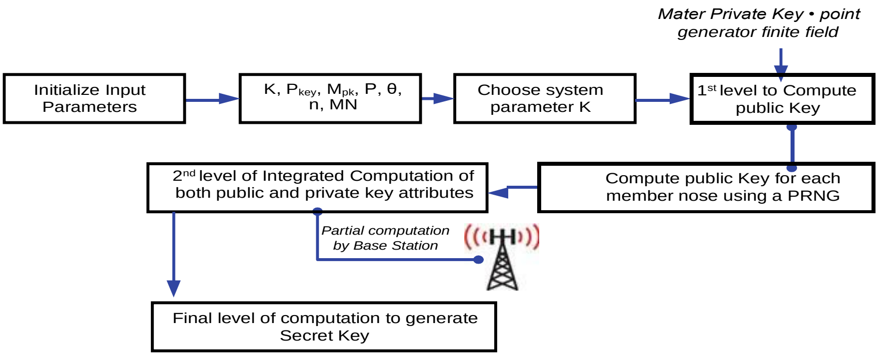


Fig. 11 FEESR key generation process

After generating the keys, FEESR constitutes another process, namely process to formulate node mobility. The computational steps which have been evaluated analytically are illustrated as follows:

Algorithm for Node Mobility

Input: S_{bound}, S_{func}, Δ

Output: New node position

Start

1. for i=1:MN
2. if (pos(MN)<S_{bound})
3. if (MN→*static*)
4. θ→2π.arb(MN)
5. θ= θ(MN)+S_{func}(PRNG)*0.1*PRNG (1,1)
6. else
7. θ= θ(MN)+(MN_{id})*0.2
8. end
9. θ=orient(θ(MN))// explain AngleWrap here
10. (x,y)→$(x,y)_{MN}$ + vel*Δ*(θ(MN))
11. End

End

The algorithm analyzes all member nodes and ensures that all node members are in the simulated boundary (step-2). If it is found to be static (step 3), the algorithm computes the direction angle, i.e., θ of any (arb) member node (Line-4). It performs to recalculation (step-5) of the θ based on signum function. Lines 7 and 9 are computed

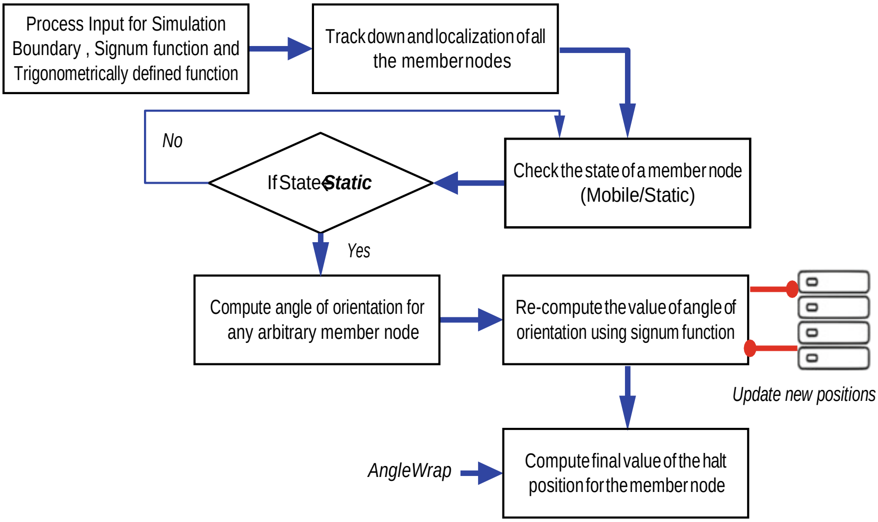


Fig. 12 FEESR algorithm for node mobility

to further upgrade new positions. The step-10 can be used to determine the final stop position of the member node. A design of the flow of this algorithm is pictorially (Fig. 24) illustrated below: This enables the sensor node highly resistant to various kinds of threats. The flow design of the algorithm is shown in the figure (Fig. 24) as follows (Fig. 12):

The study here incorporated a notion of angle wrap which is designed to formulate cost-effective node mobility. The computational steps to attain node mobility using AngleWrap which is highlighted in the (Line-10) of the above algorithm is expressed below:

6.2.1 Steps for Angle Wrap

Start

1. Initialize θ

2. If ($\theta > 2\pi$)

3. Compute $\theta = \theta - 2\pi$

4. Else if ($\theta < 0$)

5. $\theta = 2\pi + \theta$

6. End of if

7. θ=orient(θ(MN))

End

Algorithm for validating cluster key

Input: MN_{comp}, CH, c_{key}

Output: acceptance or discarding of c_{key}

Start

1. S→brd(msg(MN_{comp}))→CH, msg(CH_{comp})→MN)
2. CH→decrypt(msg(MN_{comp})), update list
3. MN→decrypt (msg(CH_{comp})), update list
4. If (key_{dec}=1)
5. Discard c_{key}
6. Else
7. Accept c_{key}

End

Suppose, If the corresponding node of the receiver gets the key which is decrypted within the cycle of transmission with member nodes (step-4), than it leaves the key of cluster and separated itself from the node that has transmitted this key. After completion of this process, it updates the nearer CH or member node immediately during its shifting toward other clusters and therefore optimum protection can be ensured. An important contribution of this algorithm is that it does not use complex key management, and the introduced system utilizes the resource benefits of the BS to exploit warning message. Therefore, the energy released for mobility is actually compensated for that verification which executed through the BS to save a lot of energy (if it is initiated from sensor side) (Fig. 13).

6.3 Feesr Numerical Analysis and Outcome Comparison

This segment illustrates the outcome accomplished after implementing the conceptualized FEESR in a numerical computing platform. The setup considers an integrated development environment to model the three different algorithmic approaches which are highlighted in the prior Sect. 3.2. This part of the study basically illustrates the significance of the outcome obtained by accessing FEESR analytical design where in the same platform performance of other improvised LEACH protocols which are highly cited till date for secure data aggregation are also simulated. The prime purpose of considering this numerical analysis is to assess the effectiveness of the FEESR by comparing it with improvised LEACH which includes energy-efficient operations at the same time also attempted to overcome the security constraints.

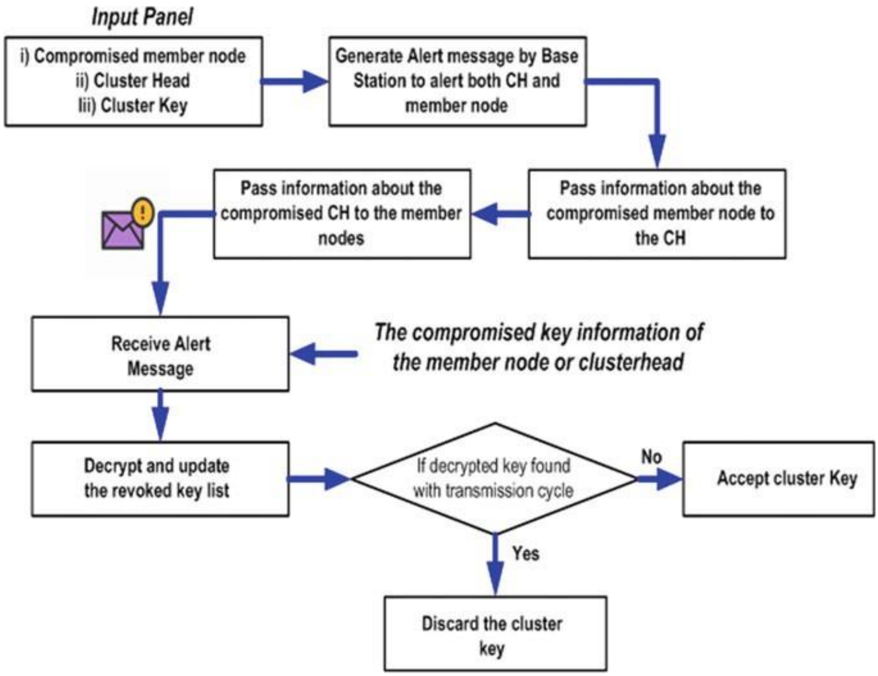


Fig. 13 FEESR lightweight cluster key management and validation by BS

In the first segment, the FEESR performance has been compared with the performance of LEACH in terms of different performance parameters and in the second segment it considers comparing the performance of FEESR with some of the standardized improvised LEACH routing protocols which are extensively discussed in the consecutive sections.

6.3.1 FEESR Comparative Performance Analysis w.r.t LEACH

The study outcomes are analyzed using 100 sensor nodes deployed in simulation area of 1200×1500 m^2. Evaluation of the results of the study was done using the complete energy utilization (Fig. 14), (ii) overall throughput (Fig. 15), and (iii) the rate of energy depleted by node mobility (Fig. 16). Compared to the existing LEACH, complete energy utilization is observed to be very low in proposed system.

- The main cause beyond this is that the introduced methodology mainly uses concept of the BS to validate the compromised nodes, which indirectly saves a huge power between a sensor node.
- FEESR system uses lightweight hashing technique, PRNG and non-repetitive encryption procedure that results in reduction of the time consumed in encryption process, that can result in some level of remaining energy stored within the node.

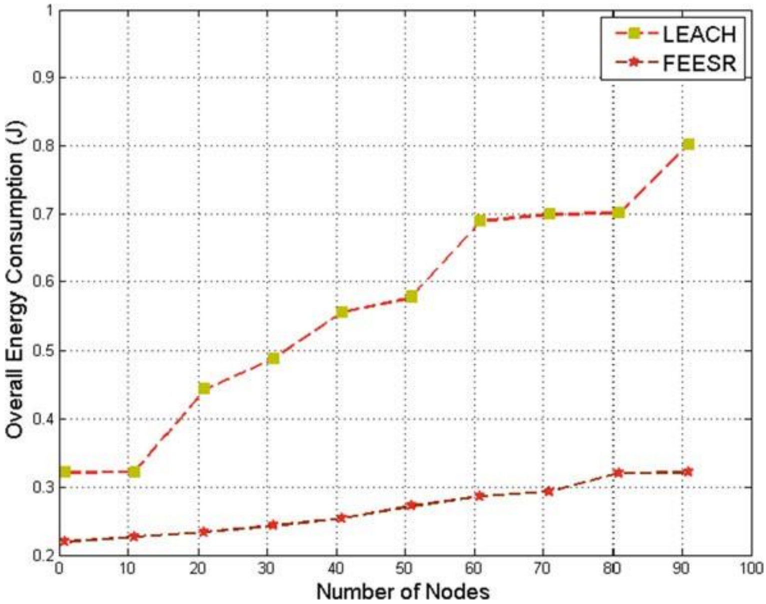


Fig. 14 Comparative analysis of energy performance in between LEACH and FEESR

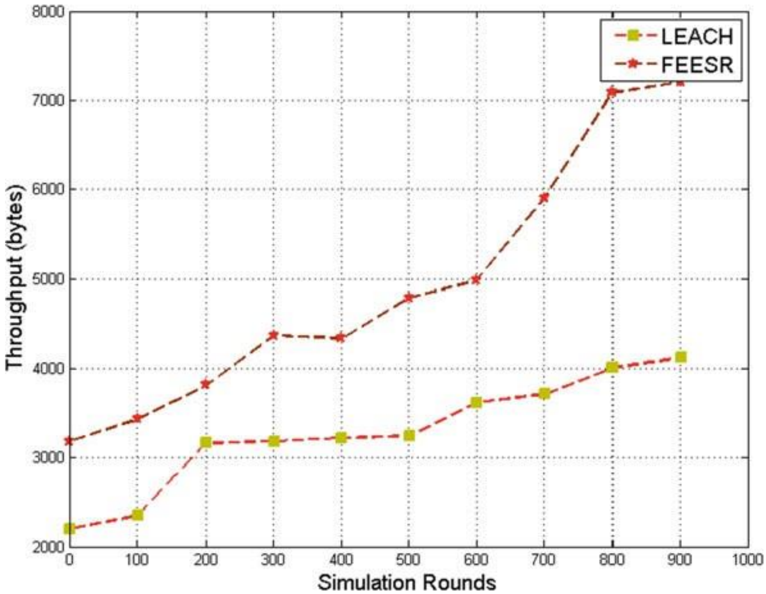


Fig. 15 FEESR Analysis of throughput as compared to LEACH

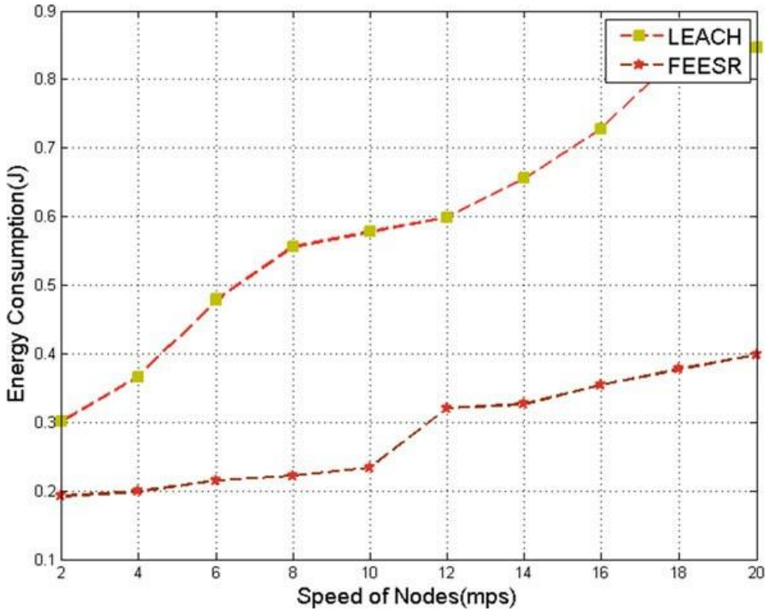


Fig. 16 FEESR analysis of energy consumption w.r.t node mobility

- FEESR adopted concept of partial key (from the BS) during generation of key process that minimizes the power which is consumed during process of key receiving by node.

Therefore, the proposed FEESR system is found to be superior for preserving approximately 50% energy consumption compared to traditional LEACH algorithms. Figure 14 also shows that LEACH performs energy consumption optimization to a limited extent whereas FEESR has improved the energy performance by optimizing its operation by employing lightweight key-based solution. The incorporation of lightweight cryptosystem has got a significant impact in the energy performance of FEESR which is reflected in the above Fig. 14.

Figure 15 demonstrates that the results of the throughput for introduced FEESR system and Traditional LEACH. The cause of low throughput for LEACH i) There are centralized positioning problems associated with CH, (ii) all nodes have to become CH, (iii) All nodes are a lack of consideration for mobility. In contrast, FEESR adopted concept of both stationary and mobility and uses a lighter encryption technique. Another fact is that all the CHs are highly connected to each other in each clustering phase, which results in reducing the possibility of re-beacon process. Apart from this, each node maintains better routing profiling with every updates, which results in the discovery of routes along with a secure node with sufficient remaining energy. Therefore, FEESR is successful in achieving approximately 30% increase in throughput.

Figure 15 also justify the fact the increasing throughput in the case of FEESR clearly shows that due to multi-stage of authentication the data integrity associated with key attributes are properly maintained in FEESR. Thereby, the number of packet drop is very lesser as compared to LEACH. The design and modeling of FEESR also achieves scalability to balance the energy and network overhead constraints. As the FEESR employ BS to perform cluster key authentication, thereby it ensures higher level of security at the same time does not compromise with the cost of communication performance.

A careful study from all the results reveals that proposed FEESR delivers a good balance among the required optimized and robust demands such as security features, quality-aware communication services and energy efficiency. The above figure also shows that FEESR achieves higher energy conservation when the mobility of the nodes is concerned within a specific deployment region. On the other hand, LEACH consumes much due to centralized positioning problem of BS. The next segment of the study further illustrated the comparative performance analysis of the FEESR with respect to standardized LEACH-based routing protocols. The observation has been carried out from 1 to 900 simulation rounds.

6.3.2 FEESR Extensive Performance Analysis w.r.t Improved Versions of LEACH

This section presents an extensive performance analysis of FEESR, for this purpose the study has taken LEACH and its associative improvised routing protocols such as SecLEACH and SLEACH. The prime reason behind taking LEACH and its standardized version is these baselines are currently highly cited and gained much attention in the domain of WSN for its potential features which handles security as well as routing aspect efficiently to a greater extent. The comparative performance analysis in this regards makes FEESR outcome justifiable and efficient with respect to different parameters such as (i) throughput, (ii) overall energy consumption, and (iii) storage cost. Figure 17 shows as compared to the existing improvised LEACH-based secure routing, FEESR accomplishes better outcome when the throughput is concerned. The analysis of throughput shows FEESR pattern of throughput curve is gradually increasing with respect to simulation rounds. Therefore, it can be observed that the potential security features adopted during design and modeling of FEESR make it defensive against the above mentioned all forms of attacks (i.e., highlighted in Table 3.1). Thereby, it results in minimization in the possibility of occurrence of packet drops in the deployed WSN. On the other hand, SecLEACH, SLEACH and LEACH achieve almost similar pattern of curve owing to their conventional routing features. The outcome thereby shows FEESR significantly achieves higher throughput than the improvised LEACH-based outing schemas.

The following Fig. 4.10 also shows that FEESR conserves a better amount of energy due to its less iterative cryptographic operations which is not found in the case of LEACH and SecLEACH.

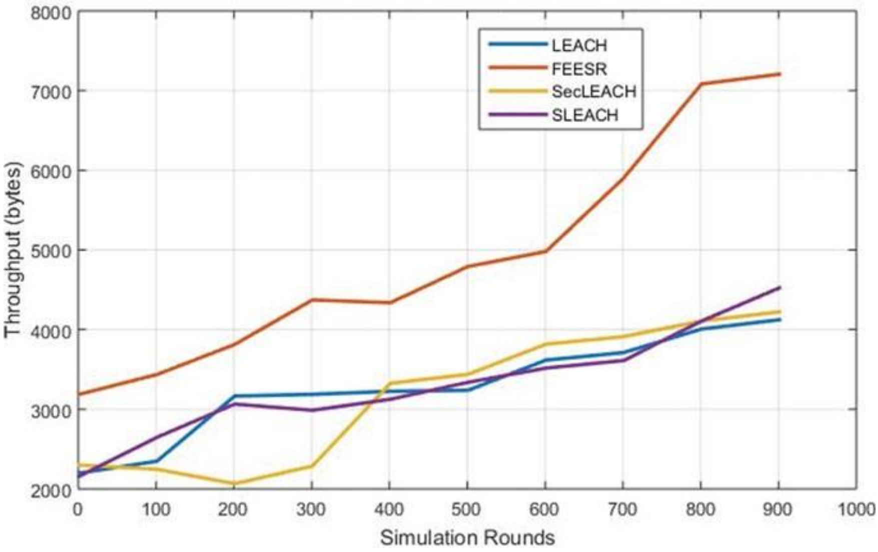


Fig. 17 FEESR comparative analysis of throughput

The outcome shows that SLEACH achieves energy consumption with an inconsistent pattern of curve.

In Figs. 18 and 19 it is clearly shown that FEESR accomplishes optimal outcome when energy consumption with respect to number of nodes and the speed of the node is considered respectively. It is also demonstrated in the previous section which FEESR outperforms the conventional baselines (Fig. 20).

The study also incorporated a storage cost analysis where it is seen, in the case of FEESR the storage (RAM) utilization is very less when the public key construction phase is assessed during the numerical simulation. It is found quite higher in the case of LEACH when the size of the public key differs between (140–180) bits of data size. The study has performed this numerical simulation with respect to the hypothetical notion of implementation which is subjected to enhance the numerical computing outcome.

6.4 Result Analysis

This segment of the study basically shows the numerical outcome obtained after performing a comparative analysis where the performance evaluation of DSP-IR has been assessed with respect to different factors. The entire simulation setup considers a numerical computing environment where the design and modeling of the algorithms being carried out. The simulation analysis basically shows how along with RSA and AES, the proposed DSP-IR performs toward achieving MAC layer security.

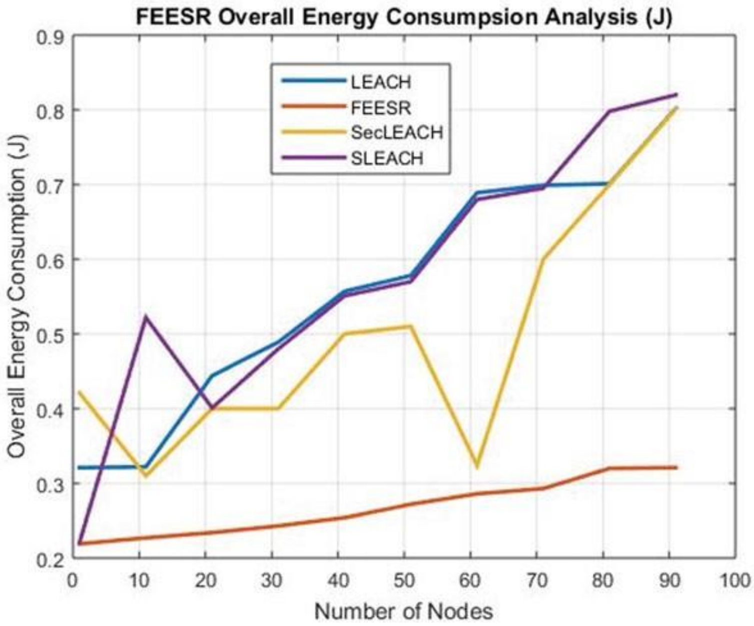


Fig. 18 FEESR analysis of energy consumption w.r.t node mobility

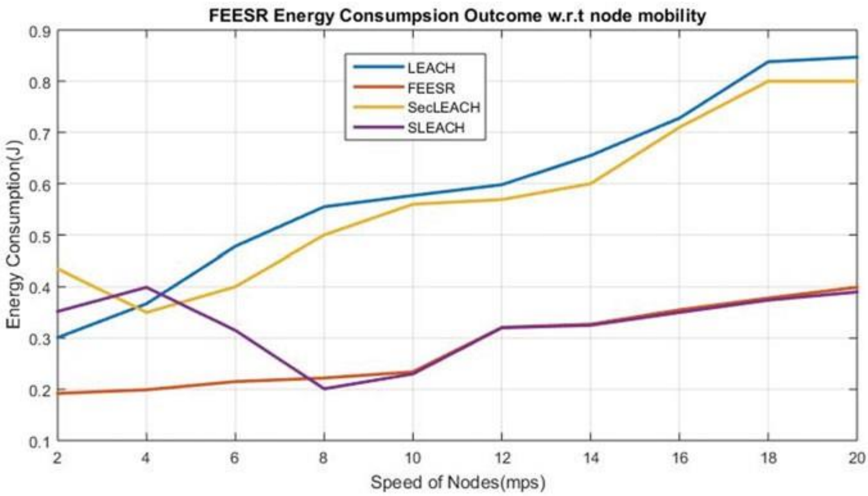


Fig. 19 FEESR analysis of energy consumption w.r.t node mobility

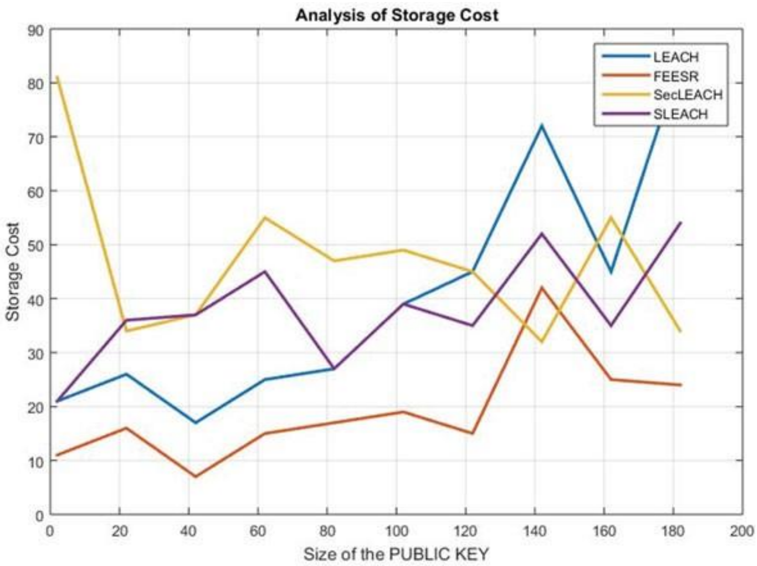


Fig. 20 FEESR analysis of storage cost w.r.t. variable length of public key size

The following Fig. 26 exhibits the analysis of average packet delay in the case of IH-MAC, S-MAC and Q-MAC when integrated with DSP-IR routing aspects.

The model description is addressed in Sect. 4.1 for many performances evaluations. (Figs. 21, 22, 23, 24 and 25). The default MAC protocol of WSNs is S-MAC,

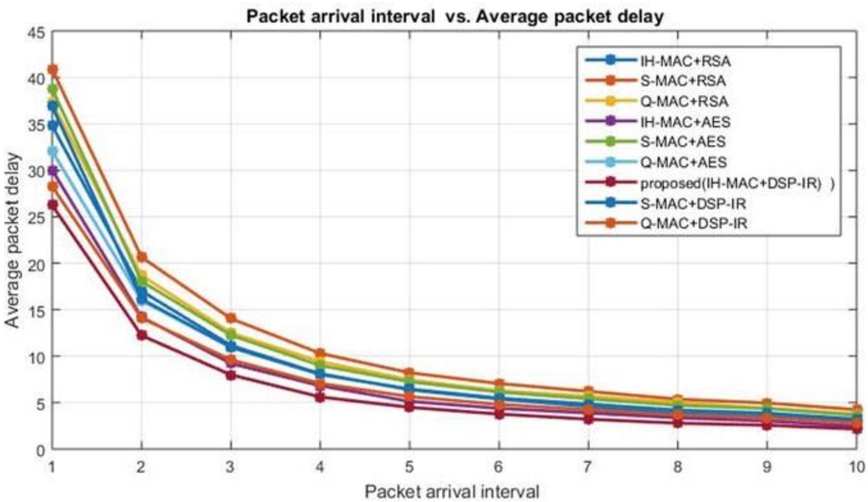


Fig. 21 Analysis of DSP-IR average packet delay (Sec)

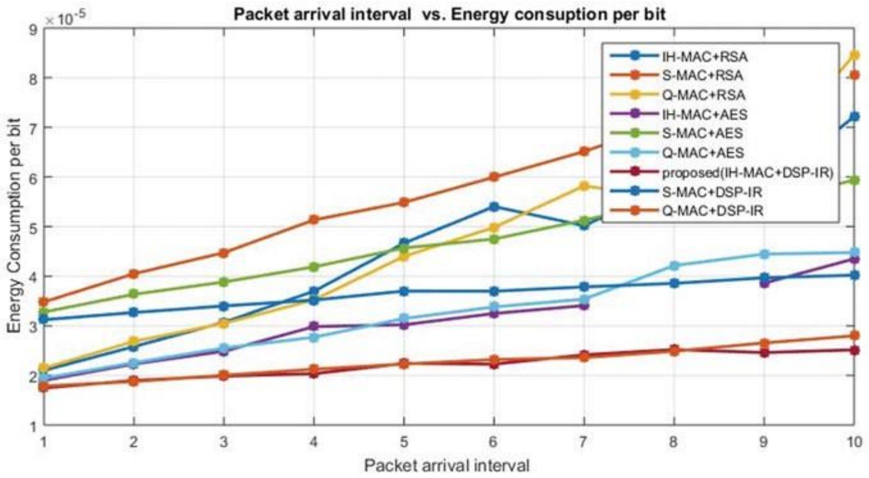


Fig. 22 Analysis of DSP-IR energy utilization per bit

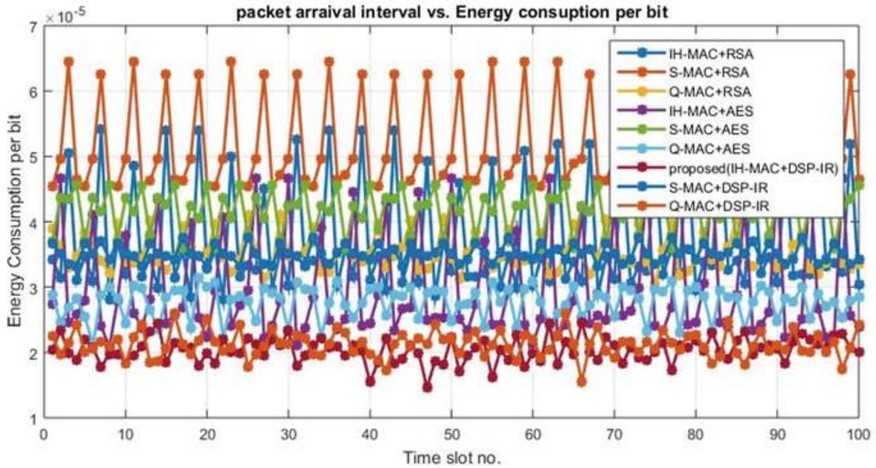


Fig. 23 Analysis of DSP-IR energy consumption per bit w.r.t time slot no

that is totally different from the traditional protocol in IEEE 802.11 because it needs energy preservation mechanism through sleep-scheduling process in both event and non-event environment. The validation of DSP-IR is carried out with AES technique S-MAC, Q-MAC RSA and IH-MAC [12, 13]. As a key QoS parameter in the WSN, there is a trade-off between complete cost-performance and communication reliability concerning quality-aware data delivery services.

Figure 21 shows due to delay sensitive routing DSP-IR accomplishes better outcome with increasing packet arrival interval when implemented with the MAC

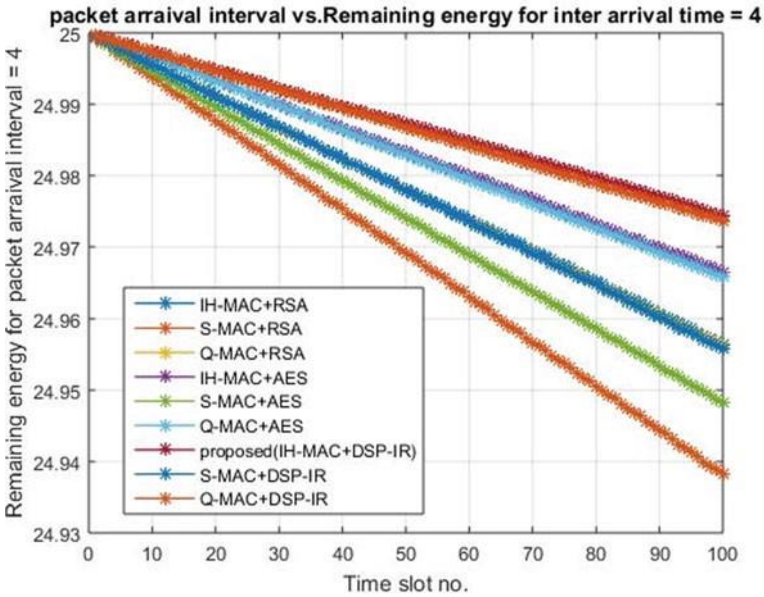


Fig. 24 Analysis of DSP-IR (Remaining energy for packet arrival interval vs. Time Slot)

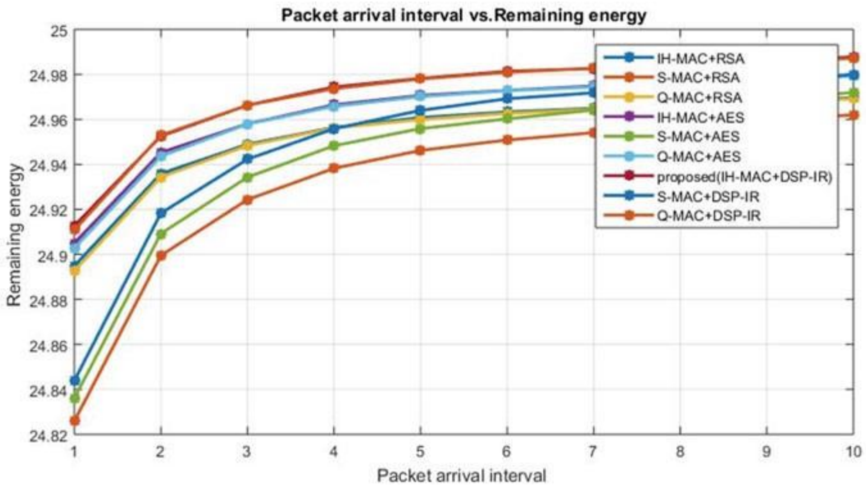


Fig. 25 Analysis of DSP-IR (Remaining energy for packet arrival interval)

layer protocols, but at the same time, RSA and AES do not provide better solution due to their highly iterative security solutions. It is also observed that DSP-IR provides better delay outcome while working with the IH-MAC solution which the protocol formulated in this study.

The structure of delay for average packet delay is lower for the introduced IH-MAC-DSP-IR protocol when considered to other combinations, therefore it is clear that it makes more suitable for energy optimization. The comparative analysis thereby justifies that the possibility of packet drop got reduced to a significant extent in the case of DSP-IR while operated with the MAC layer protocols.

The above Fig. 22 also shows that the proposed DSP-IR consumes very less energy while working with IH-MAC protocol, and at the same instance, it also achieves superior quality of security mechanism. Figure 22 reveals that trend of the energy utilization per bit against raising value of packet arrival duration and the utilization of energy per bit trend is lower for the IH- MAC -DSP-IR when it is compared to other different forms of combinations. Therefore, based on this assessment it can be stated that the proposed approach is more stable in terms of energy utilization.

Figure 23 also shows that the energy consumption in the case of DSP-IR is quite lesser and consistent when its performance is evaluated with the existing baselines. The key cause behind this is that the optimized routing operation which has influenced the energy performance to a greater extent. On the other hand, RSA consumes comparatively much more energy while being integrated with the MAC layer protocols.

Figure 24 also shows that energy conservation in the case of DSP-IR is higher while implemented with the MAC layer security in contrast with RSA and AES. The higher degree of energy conservation also ensures reliable data packet transmission while in MAC layer intelligent delay sensitive route establishment happens.

The analysis of DSP-IR also evaluated with respect to remaining energy assessment where it is observed that energy conservation in the case of proposed routing technique is quite higher and also it achieves resiliency against maximum possible threats during network intrusions. Though RSA and AES also reported to be very efficient security policies, it lacks effectiveness toward providing the balance between the communication and energy performance in a WSN (Fig. 26).

The average delay also reduced in the case of DSP-IR which implicates better energy and security performance during the route establishment operations. It also ensures that the intelligent routing takes place by considering the shortest routes where the possibility of node compromise attack ~0.

7 Conclusion

The design and modeling of the FEESR targeted to overcome all the limitations associated with the existing highly cited security approaches such as SLEACH [56], SecLEACH [57] and MS-LEACH [6] which are mostly applicable to few specific types of attacks. The security model of FEESR is designed in a way where a lightweight key-based crypto-solution is imposed which basically includes three different policies such as (i) key generation policy, (2) assessment of node mobility along with (3) a policy to validate cluster key. The complete design and implementation of FEESR are carried out in a numerical computing environment in terms

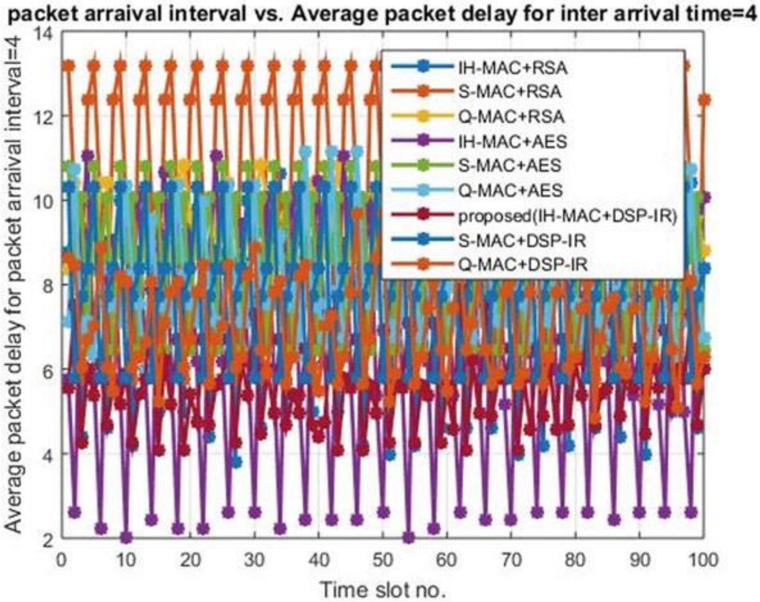


Fig. 26 Analysis of DSP-IR (Average packet delay vs. Time slot no.)

of different performance parameters such as storage cost, energy consumption, and throughput. The extensive analysis shows that FEESR outperforms LEACH and all its improvised solutions in terms of energy and security. The hypothetical numerical analysis also shows that due to lightweight modeling of cryptography modeling, FEESR establishes secure data transmission while achieving very less energy consumption. The first contributory aspect of this model is deployment of base station in security system which is not addressed in most of the existing solutions. The second contributory aspect includes mobility model of both CH and member nodes where sink is static. And finally, the third contribution is where FEESR obtained better communication and energy performance with higher supportability of spontaneous updation of keys. Another novelty of this FEESR is that the CH can easily identify the mobility status associated with each member node. The security and routing performance statistics clearly show the FEESR attains better security and energy solutions by improvising its design policy where its outcome does not impose inconsistency to the system and also provides robustness with respect to variable public key size, node mobility and simulation rounds.

The design and of DSP-IR intends to perform a secure and intelligent routing with optimized communication performance. It has basically improved the security performance in conventional MAC layer. The lightweight encryption policy optimizes the cost and path metrics to enhance the energy performance as well. The novelty of this DSP-IR solution is that it introduces a delay sensitive routing and encryption policy which has significant impact on both routing and security performance from energy viewpoint. The communication modeling is designed also by incorporating a hypothetical adversarial model which is completely based on mathematical computation. The objective function formulated here basically attempts to minimize the energy consumption with higher layer security modeling and also incorporates delay compensation during routing. The encryption policy derived in this phase includes minimal computational steps with simplified mathematical operation to generate cipher-text which is quite unique in nature as compared to the other security protocols. The experimental outcomes illustrate that DSP-IR is more compatible while operating with IH-MAC protocol and outperforms the conventional S-MAC and Q-MAC in terms of average packet delay, energy consumption per bits and remaining energy for packet arrival interval. The encryption policy also can be scaled up with different key size and along with IH-MAC it provides delay optimized routing. The overall energy consumption and bit energy utilization are also found less in the case of DSP-IR due to its low computational complexity whereas IH-MAC while evaluated with RSA and AES gives comparatively higher energy consumption.

7.1 Scope and Limitations of the Study

The scopes of the study are as follows:

- The design aspects of FEESR and DSP-IR are completely analytical and simulated over numerical computing environment with hypothetical adversarial modeling.
- The conceptual modeling and optimization modeling considers a set of specific design and operational constraints.
- Simulation carried out for a specific set of sensors where hypothetically the sensor configurations are considered. Several difficulties faced during research which includes obtaining testbed and physical nodes for hardware simulations.

7.2 Future Scope of Applicability

The future enhancement of the proposed study is anticipated as follows:

- Adaptability into different intelligent routing schema adopted for futuristic IoT-based critical applications and also ensure a strong dimension of research future.
- The computational modeling can be realized with real-time operational constraints in IoT and other sensor network applications.

References

1. Chen L, Ji J, Zhang Z (2013) Springer Science & Business Media
2. Pathan A-SK (2016) Security of self-organizing networks: MANET, WSN, WMN, VANET. CRC Press
3. Wen H (2013) Physical layer approaches for securing wireless communication systems. Springer Science & Business Media, Berlin
4. Kesavan VT, Radhakrishnan S (2016) Cluster based secure dynamic keying technique for heterogeneous mobile. Wirel Sens Netw China Commun 13(6):178–194
5. Shu T, Krunz M, Liu S (2010) Secure data collection in wireless sensor networks using randomized dispersive routes. IEEE Trans Mob Comput 9(7):941–954
6. Saleem K, Fisal N, Al-Muhtadi J (2014) Empirical studies of bio-inspired self-organized secure autonomous routing protocol. IEEE Sens J 14(7):2232–2239
7. Roy S, Conti M, Setia S, Jajodia S (2014) Secure data aggregation in wireless sensor networks: filtering out the Attacker's impact. IEEE Trans Inf Forensics Secur 9(4):681–694
8. Liu X (2015) Atypical hierarchical routing protocols for wireless sensor networks: a review. IEEE Sens J 15(10):5372–5383
9. Wei Z et al. (2018)K-CHRA: a clustering hierarchical routing algorithm for wireless rechargeable sensor networks. in IEEE Access
10. Ngangbam R, Hossain A, Shukla A (2018) An improved clustering based hierarchical protocol for extending wireless sensor network lifetime—EG LEACH. In: 2018 IEEE international conference on system, computation, automation and networking (ICSCA), Pondicherry, pp 1–5
11. Pantazis NA, Nikolidakis SA, Vergados DD (2013) Energy-efficient routing protocols in wireless sensor networks: a survey. IEEE Commun Surv Tutorials. 15(2):551–591, Second Quarter 2013
12. Chen YP, Liestman AL, Liu J (2006) A hierarchical energy-efficient framework for data aggregation in wireless sensor networks. IEEE Trans Veh Technol 55(3):789–796
13. Singh SK, Kumar P, Singh JP (2017) A survey on successors of LEACH protocol. IEEE Access 5:4298–4328
14. Heinzelman WB, Chandrakasan AP, Balakrishnan H (2002) An application-specific protocol architecture for wireless microsensor networks. IEEE Trans Wirel Commun 1(4):660–670
15. Handy MJ, Haase M, Timmermann D (2002) Low energy adaptive clustering hierarchy with deterministic cluster-head selection. In: 2002. 4th international workshop on mobile and wireless communications network, pp 368–372
16. Voigt T, Dunkels A, Alonso J, Ritter H, Schiller J (2004) Solaraware clustering in wireless sensor networks. In: Proceedings of the ISCC 2004 ninth international symposium on computers and communications, vol. 1, pp 238–243
17. Perrig A, Szewczyk R, Tygar JD, Wen V, Culler DE (2002) Spins: security protocols for sensor networks. Wirel Netw 8(5):521–534
18. Oliveira LB, Wong HC, Bern M, Dahab R, Loureiro AAF (2006) Secleach—a random key distribution solution for securing clustered sensor networks. In: Proceedings of the Fifth IEEE international symposium on network computing and applications, ser. NCA '06. IEEE Computer Society, Washington, DC, pp 145–154
19. Gnanambigai J, Rengarajan N, Anbukkarasi K (2013) Q-leach: an energy efficient cluster based routing protocol for wireless sensor networks. In: 2013 7th international conference on intelligent systems and control (ISCO), pp 359–362
20. Chen J, Shen H (2007) Meleach an energy-efficient routing protocol for wsns. Chin J Sens Actuators 9(4):035
21. Junping H, Yuhui J, Liang D (2008) A time-based cluster-head selection algorithm for leach. In: IEEE symposium on computers and communications, ISCC 2008, pp 1172–1176
22. Abuhelaleh MA, Mismar TM, Abuzneid AA (2008) Armorleach—energy efficient, secure wireless networks communication. In 2008 Proceedings of 17th international conference on computer communications and networks, pp 1–7

23. Ali MS, Dey T, Biswas R (2008) Aleach: advanced leach routing protocol for wireless microsensor networks. In: International conference on electrical and computer engineering, ICECE 2008, pp 909–914
24. Hong J, Kook J, Lee S, Kwon D, Yi S (2008) T-leach: the method of threshold-based cluster head replacement for wireless sensor networks. Inf Syst Front 11(5):513–521
25. Wang W, Wang Q, Luo W, Sheng M, Wu W, Hao L (2009) Leach-h: an improved routing protocol for collaborative sensing networks. In: International conference on wireless communications signal processing, WCSP 2009, pp 1–5
26. Ren P, Qian J, Li L, Zhao Z, Li X (2010) Unequal clustering scheme based leach for wireless sensor networks. In: 2010 fourth international conference on genetic and evolutionary computing (ICGEC), pp 90–93
27. Tong M, Tang M (2010) Leach-b: an improved leach protocol for wireless sensor network. In: 2010 6th international conference on wireless communications networking and mobile computing (WiCOM), pp 1–4
28. Liu J-L, Ravishankar CV (2011) Leach-ga: Genetic algorithmbased energy-efficient adaptive clustering protocol for wireless sensor networks. Int J Mach Learn Comput 1(1):79–85
29. Al-Ma'aqbeh F, Banimelhem O, Taqieddin E, Awad F, Mowafi M (2012) Fuzzy logic based energy efficient adaptive clustering protocol. In: Proceedings of the 3rd international conference on information and communication systems, ser. ICICS '12. ACM, New York, pp 21:1–21:5
30. Wang A, Yang D, Sun D (2012) A clustering algorithm based on energy information and cluster heads expectation for wireless sensor networks. Comput Electr Eng 38(3):662–671. the Design and Analysis of Wireless Systems and Emerging Computing Architectures and Systems
31. Xiao M, Zhang X, Dong Y (2013) An effective routing protocol for energy harvesting wireless sensor networks. In: 2013 IEEE wireless communications and networking conference (WCNC), pp 2080–2084
32. Beiranvand Z, Patooghy A, Fazeli M (2013) I-leach: an efficient routing algorithm to improve performance amp; to reduce energy consumption in wireless sensor networks. In: 2013 5th conference on information and knowledge technology (IKT), pp 13–18
33. Mahmood D, Javaid N, Mahmood S, Qureshi S, Memon AM, Zaman T (2013) Modleach: a variant of leach for wsns. In: 2013 Eighth International Conference on broadband and wireless computing, communication and applications (BWCCA), pp 158–163
34. Abdulsalam HM, Kamel LK (2010) W-leach: weighted low energy adaptive clustering hierarchy aggregation algorithm for data streams in wireless sensor networks. In: 2010 IEEE international conference on data mining workshops, pp 1–8
35. Chen H, Zhang C, Zong X, Wang C (2013) Leach-g: an optimal cluster-heads selection algorithm based on leach. J Softw 8(10):2660–2666
36. Bsoul M, Al-Khasawneh A, Abdallah AE, Abdallah EE, Obeidat I (2013) An energy-efficient threshold-based clustering protocol for wireless sensor networks. Wirel Pers Commun 70(1):99–112
37. Tripathi M, Battula RB, Gaur MS, Laxmi V (2013) Energy efficient clustered routing for wireless sensor network. In: 2013 IEEE ninth international conference on mobile Ad-hoc and sensor networks (MSN), pp 330–335
38. Chouikhi S, Korbi IE, Ghamri-Doudane Y, Saidane LA (2015) A survey on fault tolerance in small and large scale wireless sensor networks. Comput Commun 69:22–37
39. Salim A, Osamy W, Khedr AM (2014) Ibleach: intra-balanced leach protocol for wireless sensor networks. Wireless Netw 20(6):1515–1525
40. Eletreby RM, Elsayed HM, Khairy MM (2014) Cogleach: a spectrum aware clustering protocol for cognitive radio sensor networks. In: 2014 9th international conference on cognitive radio oriented wireless networks and communications (CROWNCOM), pp 179–184
41. Sasikala SDA, Sangameswaran N (2015) Improving the energy efficiency of leach protocol using vch in wireless sensor network. Int J Eng Dev Res 3(2):918–924
42. Tang C, Tan Q, Han Y, An W, Li H, Tang H (2016) An energy harvesting aware routing algorithm for hierarchical clustering wireless sensor networks. KSII Trans Internet Inf Syst (TIIS) 2(2). [Online]. Available: https://doi.org/10.3837/tiis.2016.02.004

43. Batra PK, Kant K (2016) Leach-mac: a new cluster head selection algorithm for wireless sensor networks. Wirel Netw 22(1):49–60
44. Ding M, Cheng X, Xue G (2003) Aggregation tree construction in sensor networks. In: 2003 IEEE 58th vehicular technology conference, pp 2168–2172
45. Kim H, Han K (2005) A power efficient routing protocol based on balanced tree in wireless sensor networks. In: DFMA '05 proceedings of the first international conference on distributed frameworks for multimedia applications, Besançon, France, pp 138–143
46. Tan HÖ, Körpeoğlu Í (2003) Power efficient data gathering and aggregation in wireless sensor networks. ACM SIGMOD Rec 32(4):66–71
47. Qiu W, Skafidas E, Hao P (2009) Enhanced tree routing for wireless sensor networks. Ad Hoc Netw 7(3):638–650
48. Liu X, Zhang H, Xiang Q, Che X, Ju X (2013) Taming uncertainties in real-time routing for wireless networked sensing and control. IEEE Trans Smart Grid 4(1):288–301
49. Xue Y, Ramamurthy B, Vuran MC (2011) SDRCS: a service-differentiated real-time communication scheme for event sensing in wireless sensor networks. Comput Netw 55:3287–3302
50. Felemban E, Lee C-G, Ekici E, Boder R, Vural S (2005) Probabilistic QoS guarantee in reliability and timeliness domains in wireless sensor networks. In: Proceeding of the IEEE INFOCOM
51. He T, Stankovic J, Lu C, Abdelzaher T (2003) SPEED: a stateless protocol for real-time communication in sensor networks. In: Proceeding of the IEEE ICDCS
52. Li H, Cheng Y, Zhou C (2009) Minimizing end-to-end delay: a novel routing metric for multi-radio wireless mesh networks. In: Proceeding of the IEEE INFOCOM
53. Yin S, Xiong Y, Zhang Q, Lin X (2006) Traffic-aware routing for real-time communications in wireless multi-hop networks. Wirel Commun Mobile Comput 6:825–843
54. Spachos P, Hantzinakos D (2014) Scalable dynamic routing protocol for cognitive radio sensor networks. IEEE Sens J 14(7):2257–2266
55. Tunca C, Isik S, Donmez MY, Ersoy C (2014) Distributed mobile sink routing for wireless sensor networks: a survey. IEEE Commun Surv Tutorials 16(2):877–897
56. Srinivasan SPT, Chellappan C (2011) Semi-randomised propagation for secure routing in Wireless Sensor Networks. In: 2011 International conference on recent trends in information technology (ICRTIT), Chennai, pp 428–432
57. Ranjani S, Kavitha C (2014) Secured data integrity routing for Wireless Sensor Networks. In: 2014 international conference on advances in electronics, computers and communications (ICAECC). Bangalore, pp 1–6

Futuristic Communication Systems Using Mobile Edge Computing

Maninder Jeet Kaur, Piyush Maheshwari, Sadia Riaz, and Arif Mushtaq

Abstract Wireless communication networks and technologies have transformed each and every aspect of our lives. The computer-intensive applications and the Internet of Things (IoT) are the primary driving force behind the unprecedented and exponential growth of network traffic in recent times. It has also generated an enormous demand for computational requirements to meet the ever-growing and anticipated evolution of 5G networks. However, it is pertinent to highlight that the end-users, despite all this, continue to struggle with small storage-capacities and limited processing speed. To address this issue, Mobile Edge Computing (MEC) is the way forward, as it is the technology supporting the evolution of the 5G communication networks. Since it can host intensive computing applications, it creates efficient optimization of mobile resources. Before transmitting it to the cloud, it can also process extensive data and provide mobile users near proximity to cloud computing capabilities within the Radio Access Network (RAN). It offers context-aware services through the deployment of RAN knowledge. MEC, therefore, makes a wide range of applications where real-time response is appropriate, e.g. driverless cars, virtual reality, robotics and live media. Non-orthogonal multiple access (NOMA), dense heterogeneous networks (HetNets), cloud radio access networks (C-RAN), unmanned aerial vehicles (UAV), IoT, wireless power transfer (WPT) and

M. J. Kaur
Department of Applied Science and Engineering Technology, Seneca College, Toronto, Canada
e-mail: maninder.kaur@senecacollege.ca

P. Maheshwari (✉)
Faculty of Engineering and IT, The British University, Dubai, United Arab Emirates
e-mail: piyush.maheshwari@buid.ac.ae

S. Riaz
Faculty of Information Technology and Research Methods, S. P. Jain School of Global Management, Dubai, United Arab Emirates
e-mail: sadia.riaz@spjain.org

A. Mushtaq
Faculty of Management Information System, City University College of Ajman, Ajman, United Arab Emirates
e-mail: m.arif@cuca.ae

R. Tiwari et al. (eds.), *Energy Conservation Solutions for Fog-Edge Computing Paradigms*, Lecture Notes on Data Engineering and Communications Technologies 74,
https://doi.org/10.1007/978-981-16-3448-2_13

energy harvesting (EH) and machine learning (ML) are all main technologies for 5G. The MEC summary, including the fundamental characteristics, challenges and market factors, is presented in this chapter. In addition to the integration of MEC with the upcoming 5G and beyond technologies including NOMA, WPT EH, UAV, IoT and heterogeneous CRAN and ML, we also address the role of MEC in the 5G network architecture. In doing so, we address the state of the art research activities and describe the potential of MEC's future course.

Keywords Mobile edge computing · Cloud computing · Internet of things · CRAN · Computational offloading

1 Introduction

In an attempt to achieve ubiquitous computing and seamless connectivity, there have been extensive research and development efforts in the field of 5G (fifth generation) domain. Since the use of mobile technology has seen tremendous growth over the last few years, there are more and more connected devices. IT is required by the 5G wireless communications to provide high data rate (Gbps), high base station capacity and extremely low latency to have better QoS (Quality of service) experience by the users. This necessity has led to efforts to work towards the next generation of mobile communication, i.e. the fifth generation (5G), and to identify, improve and standardize systems and services for the next generation [1–3]. Initially, the Internet was built to allow communication and sharing of resources between end-to-end hosts. Over time, with technical advancements including broadband and mobile devices, the reach of the Internet has expanded. Huge amounts of content are now searched for and uploaded every day through platforms such as YouTube, Facebook, Flicker and Google. This content is rising exponentially, and it is expected that multimedia content will account for most Internet traffic in the near future. Global mobile data traffic increased by 63 percent in 2016 and is projected to hit 49 exabytes by 2021, according to the authors the report [4]. Authors in [5], mentions how millions of cars are already connected via fourth generation (4G) cellular access, and cellular connectivity to the Internet of Things (IoT) is expected to expand significantly by 2024 [6, 7]. Therefore, different systems, such as content delivery networks (CDNs) and peer-to-peer (P2P) networks, have been introduced for content distribution. As developing model for the deployment of resources at the network edge, Edge computing (EC) has also been proposed, which has fascinated considerable interest; in particular, EC brings computation, bandwidth and storage resources nearer to the end user to minimalize backbone network traffic and latency of response which makes huge difference in the processing times. However, given all this knowledge on 5G development, the deployment is still elusive of the fact that the most demanding part in 5G will be the integration and management of multiple technologies which would be required to operate and optimize in the same environment for immense number of use cases.

5G is generating a wealth of new services and mobile apps, augmented reality, processing of natural languages, immersive gaming, smart health care, etc. In comparison with cloud-native applications, these applications and services would be optimized to leverage the processing and storage benefits offered by the edge framework. Meanwhile, such 5G networks and applications are demanding for more bandwidth and computational power to satisfy their users' stringent Quality-of-Service (QoS) specifications. Mobile Edge Computing (MEC) has emerged in the 5G age as a modern and powerful technology to enable a new breed of time-sensitive apps. It addresses the existing centralized layout issue of cloud networks by deploying small computing infrastructures at the edges of the network (i.e. high end to end communication latency between the user equipment and the cloud). As a new model, MEC offers IT and cloud computing capabilities in close proximity to mobile subscribers inside the Radio Access Network (RAN) [8–10]. This is also a potential way of overcoming the battery barrier and allowing mobile devices to discharge their most energy-consuming (computational-intensive) tasks to the MEC environment whenever possible and convenient. This idea would decrease the lengthy execution period on mobile devices, resulting in a reduction in power consumption.

2 Outline of MEC and 5G

Mobile Edge Computing (MEC) offers computing resources that are in close proximity to mobile subscribers within the access network. Thanks to this proximity to mobile users, it provides a service environment with ultralow latency and high bandwidth as well as direct access to real-time radio network information that can be used by applications and providers to deliver context-related services. MEC enables acceleration of material, services and applications, increasing reactivity from the edge. The experience of mobile subscribers can be improved by effective network and service operations based on in-sight conditions for the radio and network. 5G is a new technology for wireless networking, linking billions of people, computers and devices. With very low latency, it makes large data bandwidth rates. During this process, tasks are offloaded from MEC to MEC servers. The servers are available at the end of the network edge. MEC servers' attribute is that they can share actual proximity with end-users, hence the entire MEC network is efficient in terms of energy consumption and providing low latency to the users [11–14]. Even though cloud computing models are more conventional and prevalent; they have a limitation due to QoS specifications. For that reason, Edge computing is more relevant than the cloud computing model for being responsive to highly interactive, computer-interactive applications. It also includes low latency and high throughput, which is an added advantage. Another significant limitation of the cloud computing model is that it is not energy consumption efficient due to being far from the UE (User Equipment). To be more precise, the cloud servers are usually located on the core network, while the mini-cloud edge servers are hosted on the network's edge [15]. MEC offers storage and processing facilities at the edge of the network, such as radio access networks (RANs) and base stations (BS). It also enhances context awareness and reduces latency. MEC servers deploy virtualized interface to access storage and computer facilities while being localized and several hosts (such as BSs). By collecting and providing real-time information on the services offered by each host, the available resources (e.g., network capacity and load), the network topology, as well as the management of MEC applications, a MEC orchestrator overlooks the MEC hosts [16].

2.1 Fundamentals of MEC

Mobile edge computing offers cloud computing services that are in close proximity to mobile customers within the access network. As a result, it provides an ultra-low latency and high bandwidth service environment, as well as direct access to real-time radio network information that can be used by applications and utilities to provide context-related services. Traditional operators of telecom networks execute traffic control flows (for packet forwarding and filtering), but cloud servers are also installed at each base station in the MEC. Network providers are also highly responsible for servicing mobile subscribers. It is possible to describe Mobile Edge Computing as:

"Mobile Edge Computing is a model for enabling business-oriented, cloud computing platforms to serve delay-sensitive, context-aware applications within the radio access network close to mobile subscribers." It provides software developers and content developers with real-time RAN knowledge that is used to provide mobile customers with context-aware services, enriching the loyalty of consumers and enhancing the quality of experience (QoE). MEC raises edge accountability and makes it possible to host computing and server services on the edge, reducing subscribers' network latency and bandwidth usage. New applications and new advanced services can be enabled to be quickly brought to the mobile subscribers and enterprises by the network operators [17]. The following parts would address existing MEC application scenarios such as dynamic content placement, IoT flow computing, mobile big data analytics and smart transport. As apps are computational intensive and need enormous storage space, these applications are not ideal for running on mobile or portable devices.

2.1.1 Dynamic Content Placement

Ultra High Definition (UHD) is one example of rich multimedia infotainment that is accessed content and places strict requirements over the network. Reports suggest that IP traffic(video) accounts for 80% of overall traffic in 2019 [18]. Moreover, nearly 4/5th of mobile data traffic will be generated by mobile video in 2022 [19]. Hence, it is imperative to have grounding solution such as CDNs (Content Delivery Networks) which are being used by common services (e.g. Playstation, Microsoft One Drive, Netflix, etc.) that have distributed caches to reduce the access time due to the nearness to the end user. It is suggested that edge-based content optimization will increase the performance of the network, hence increasing the quality of experience and service to the user.

2.1.2 Computational Offloading in IoT

Computing offloading does not characteristically mean that all program execution is passed to the Edge Cloud. However, a program is initially broken down into modules, some of which need to be run locally (on a mobile device) [20]. All the modules that control the peripherals for input/output. A judgment on what is more suitable to offload must be made for the remainder of the modules. So, how to break the software for offloading and how to make a decision? Any other problems will be: how to break an IoT application? How to split it? How does an IoT application define a task? How can a job be offloaded or not identified? How do I synchronize the app when the user is in motion? Answer to these questions will be eventually be discussed in this chapter.

2.1.3 Mobile Big Data Analytics

The primary way to go online in recent times is through smartphones. In that context, mobile devices are the most productive way of reaching out to target audiences for business promotion or advertising [21]. The tremendous volumes of structured and unstructured data that is generated online are called Big Data. It raises new challenges, issues, and research domains to clean, interpret and visualize the same. Likewise, data collection occurs at the core network from the edge computing devices. It is transmitted to the centralized core network to apply Big Data Analytics and further processing [17, 22].

2.2 Integration of MEC with 5G Systems

The solutions provided by the fifth-generation mobile networks (5G) provide avenues that surpass radio link capacity upgrades, deployment of new enablers to network operators and service providers. It can make the network more adaptable, yet dynamically adjusting to the evolving needs of running services. Network Function Virtualization (NFV), Software-Defined Networking (SDN) and Multi-access Edge Computing are examples of such enablers [23].

2.2.1 NFV

In commercial off-the-shelf (COTS) hardware organized into cloud-managed infrastructure networks, NFV enables various network structures to be instantiated as virtual functions.

2.2.2 SDN

Although allowing quick network (re)adjustment at the scale of individual data flows, SDN decouples the control and data planes.

2.2.3 MEC

By positioning the 5G enablers at the edge of the network and closer to end-user users, MEC addresses the requirements of latency-dependent applications, allowing for higher performance. We argue, however, that this versatility should be paired with contextual data on user mobility and content consumed, enabling network control authorities to initiate dynamic changes to both the network and the service consumed.

3 Overview of MEC/5G Researches

3.1 Internet of Things (IoT) Leveraging MEC

The integration of the Internet of Things with MEC may lead to controlled computing opportunities and capacities. However, it can discharge computation-intensive tasks and delay critical functions to the network's edge. By doing so, it produces an optimal quality of service. Generally speaking, the essence of cloud computing lies in the fact that it acts as a middle tier between a third party who wishes to avoid self-management and care of IoT data storage and would like to not compromise on computing and functionality aspects. Nevertheless, could computing also poses some concerns due to its centralized design that are primarily running the conventional cloud servers. For instance, lack of location, connectivity, latencies associated with WANs and single point of failure are some of the reported literature problems [24]. On the other front, when operating on decentralized systems, it may require mobility management, location intelligence and awareness, scalable resources, geo-distribution, low latency and ability to serve IoT applications. The use cases of IoT are instead mission-critical communication protocols that include latency as low as 1 ms and reliability as high as 99.99 per cent. To elaborate this, let us assume how a factory automation application may need packet loss reliability of 9–10 packets, while latency range from 250 µs to 10 ms [25, 26]. Therefore, several shortcomings and susceptibilities will be exposed due to the integration of IoT applications and centralized cloud servers. And looking at the exponential growth rate of IoT devices recently, and the velocity of Big Data produced, many network traffic and communication challenges will eventually come to limelight.

The exploitation of MEC for IoT applications is primarily realized by assessing technical parameters, such as resource allocation, computation offloading, scalability, connectivity, privacy and protection. However, this integration will make an enormous contribution towards the 5G Vision, which is widely claimed to be the ultimate game-changer, as far as futuristic mobile network generations are concerned. In that context, IoT is considered a critical use case for MEC by the proponent of MEC, a technology that has only recently been developed. The MEC server is the platform or gateway between massive IoT networks, vital latency and core network that provides edge-computing services and functionalities. With MEC technology, IoT application domains are empowered by extending some knowledge to the edge of the network. Although MEC would provide IoT networks with on-site cloud storage services, there are still challenges in terms of heterogeneity, scalability, mobility, and protection of devices and networks. Table 1 provides a list of major IoT-relevant MEC surveys.

Table 1 Summary of important surveys on MEC research relevant to IoT

Feature	MEC research	Relevance to IoT
Classification	MEC's taxonomy is focused on numerous factors, including its functionality, access technologies, applications and objectives [10]	Classifies MEC applications as offloading computations, Collaborative computation, IoT replication of memory and distribution of information
	Classification of deployed applications [27]	Does not mention IoT
MEC-IOT use cases	Overview of MEC's position in IoT applications [28]	Implementations for IoT cases are provided: security, security, and data analytics, vehicular comm, edge cloud comm
Architecture and computation offloading	MEC standardization efforts [29]	This describes MEC as an IoT gateway
Virtualization	Other technologies like VMs, SDN, etc. survey and overview [30]	It describes how a local IoT gateway feature capable of conducting data aggregation and big data analytics for application domains can be included in the MEC framework

(continued)

Table 1 (continued)

Feature	MEC research	Relevance to IoT
	An investigation into how SDN can be utilized to allow edge Computing [31]	Discuss SDN scenarios and future studies focused on IoT and edge computing
	Network slicing elaboration from the E2E [32]	IoT with 5G
Security	Threats to security and problems in the Edge paradigms [33]	No explicit IoT debate
Communication	A systematic survey on shared control of radio and computing resources in MEC systems [34]	The role of MEC in the IoT is introduced briefly
	A systematic survey of computing, caching and communication strategies problems in the MEC [35]	IoT use scenarios, including healthcare, wireless sensor systems, smart grid, smart house, and smart city
Research directions	Scientific and technical advances specific to the MEC area [36]	Identifies the IoT big-data analytics MEC providers
	Three Edge Computing Technologies Succinct Tutorial, MEC, cloudlets, and fog computing included [37]	Describes the use of IoT edge computing technologies with respect to efforts, concepts, architectures, and implementations for standardization
	Concepts, implementations, opportunities and research challenges associated with MEC [38]	It does not have a detailed IoT definition. Identifies the handling of IoT data as a primary MEC case for usage
	Advanced and future MEC research directions [39]	Concise explanation of how MEC can increase latency
	Developing an edge-centric vision and its future challenges in research [40]	No explicit IoT emphasis

3.2 MEC with NOMA

With an ever-increasing exponential growth experienced in the number of IoT devices, multiple threats and challenges are faced by the more conventional orthogonal multiple access (OMA) schemes. Its biggest challenge comes in the form of meeting high link density requirements. Therefore, Non-orthogonal multiple access (NOMA) technique [41–46] and networks operating on it have outperformed other numerous access schemes based on sum capacity. NOMA technique can attain huge links, while also systematically increasing the spectrum's performance and minimizing the latency issues. NOMA's primary concept is to integrate multiple users who can transmit data simultaneously on the same resource block. Any successive cancellation of interference (SIC) is applied only to reduce co-channel interference and decoding of the target signal. It acts as an efficient multi-user detection technique at the end of the receiver. Owing to the amazing benefits offered by MEC and NOMA, the current research emphasis is on the integration of two concepts in the framework of wireless networks to yield higher efficiency and resource usage optimization [47–50]. In the study reported [47], authors have suggested developing heuristics algorithms for single-cell MEC-aware NOMA. The authors have emphasized that computing resources, consumer clusters and transmitting powers integration in the framework to reduce energy consumption. In [48], however, the authors have developed a joint SIC ordering and a scheme that deploys MEC-aware NOMA NB-IoT network to minimize the overall execution delays for each bit. Likewise, authors in [49] had devised two different user scenarios, one with single and the other with multiple users. The study examines and optimizes offloading workloads along with transmission time to reduce the delay while meeting the computing requirements of each user. In [50], the authors proposed a scheduling-based algorithm for the device-to-device (D2D)-assisted and NOMA-based MEC method. In D2D user collaboration was given a boost to examine the performance impact of the edge server's computing burden. It was also jointly optimized for allocation of computing resources, power and channels to reduce system costs.

3.3 MEC with Heterogeneous CRAN

The primary objective of MEC 5G network is to act as a platform between people, computer and service connections. The Radio Access Network (RAN) depends upon a scalable system to realize the optimal output of the 5G network for incremental promotion and usage. In this regard, a key technology is 5G supported Cloud Radio Access Networks (C-RAN) that provides unprecedented time delays and energy consumption [51]. The difference between conventional RAN and the innovative CRAN is that the later has is designed with independent baseband units (BBU) and Remote Radio Heads (RRH). The central BBU pool receives all processors, while the distributed RRHs function is to forward received radio signals from users to the BBU through front haul association. In that case, basic level transmission functions are required to be accomplished by RRHs that immensely reduce the costs related to design, operation and implementation of highly-populated and dense networks. Everyone realizes the importance of cloud computing in recent times, and how it has been widely adopted by so many users having access to a shared pool of resources [52, 53]. Since the gap between cloud and the terminal computer is usually immense, cloud computing services suffer in terms of providing low-latency across various networks and applications. That is the reason, Mobile edge computing (MEC) is explored to understand how computing resources can be utilized through optimization and deliver high service quality [QoS] closer to the terminal of end-users who require dense computing and low latency [54]. For this particular reason, computing offloading and distribution of resources has received massive attention to be among the use cases of the MEC framework [55]. Additionally, some recent research suggests MEC offloading, and resource allocation is essential considerations for design objectives [56]. Some studies also suggested a binary optimization calculation method that uses stream optimization to handle the inconsistency of data rate in wireless communications. This method is useful for convex optimization. One of the studies reviewed discusses the relationship between C-RAN communication and computation from two physical resources and time dimensions [57]. The investigation can substantiate the significance of rational resource allocation in the BBU pool. The paper [58], however, discusses the in-depth state of the channel. Given that wireless channels go through a high randomness level, studies reviewed may not be appropriate as they are not based on dynamic systems. Many researchers and studies are available in highly dynamic and complex environments that propose distributed solutions using game theory. The study reported [59] examines distributed computing's discharge mechanism based on game theory [59]. It obtains a lower system model accurately, but only in limited conditions, however, this hypothesis is always unreasonable.

3.4 MEC with UAV Communications

As for meeting on-demand deployment of wireless networks, Unmanned Aerial Vehicles (UAVs) are the most trustworthy solution due to their agility [60]. There is devoted research in this domain, particularly in the UAE-enabled base stations that provide insightful information. One such report discussed how it helped maximizing the number of land users, served by the UAV base station, subject to quality of service (QoS) constraints [61]. There have been proposals to minimize the outage by jointly optimizing the UAV trajectory and jammer power [62]. The minimum average secrecy rate was maximized in [63, 64] by jointly optimizing the trajectories and transmitting powers of both the time-division multiple access (TDMA) and frequency division multiple access (FDMA) UAV base station and UAV jammer. In [65], the authors suggested an effective iterative algorithm to optimize the worst-case secrecy potential by considering the location uncertainty of the eavesdropper.

3.5 MEC with WPT and EH

A plethora of research is available that concentrates on investigating and exploring possible solutions to deliver wireless electricity, without going through the hassle of linking cables. Mr Nikola Tesla, the visionary and acclaimed inventor, had suggested to transmit energy into the free space and converting wireless energy into direct current electricity. This stance and vision have opened so many research and futuristic avenues to develop new methods of power supply, including the very recent energy harvesting (EH) as well as wireless power transfer (WPT). The research outcome will produce fast processors, excellent connectivity and bright displays; all using limitless wireless capacity. Further, sustainability is ensured through energy recycling and eliminating wasted energy from the atmosphere. It is becoming evident as research in this domain is gaining momentum and maturity that it is possible to use renewable energy sources. The sources, like solar, sound, motion, wind and electromagnetic radiations, are environmentally friendly and more sustainable ways to generate power for low-powered devices. However, for RFIDs, power sensors, biomedical implants, wearable technologies, EH seems the optimistic and realistic solution. It is a misconception, that whatever energy will be extracted through renewable sources, will be insufficient to meet the requirements of computer-intensive tasks. It is only becoming evident with the advent of silicon technology, that even a small amount of energy produced will be able to accomplish many tasks, activities and work. Further, as we will move on to build energy storages, fewer and fewer systems can be built for batteries to reduce the harmful environmental hazards of battery output and discharge. The cost of charging and recharging batteries, particularly large in number, is also extremely high [66] and only escalates in large-scale wireless sensor networks (WSN). The solar energy thus can be used to generate battery-less WSNs through harvesting methods, using photovoltaic cells. In this case, the life of the device is not restricted by the capacity of the battery, and the operating expenditures are also reduced in the process. While discussing the sources, another attention-grabbing source is kinetic motion. An electro-mechanical energy converter may use the action of pressing a switch to broadcast a telegram to power a light. Because of the dense distribution of WiFi access points and cellular base stations [67], ambient RF radiation is an omnipresent power source in urban areas. Wireless power transmission (WPT) [68] and mobile edge computing (MEC) [69] are two promising technologies for extending the life of the device and improving the computing capability of the device, respectively. WPT's main idea is to allow the energy source, e.g. the power beacon, base station, to charge the IoT devices through microwave irradiation on demand. For example, the authors in [70] considered a cognitive machine-to-machine (M2M) communication system based on energy harvesting (EH) under laying a single-cell cellular network, where multiple M2M transmitters harvest energy from ambient radio frequency (RF) signals and investigate the problem of maximization of energy efficiency (EE). The IoT devices can offload their partial tasks to nearby MEC servers with more computing resources when in MEC, so that the tasks can be processed successfully within the delay budget. However, the energy-constrained and computing-limited problems in IoT systems can only be solved simultaneously by using WPT or MEC, and this inspired us to integrate the above two advanced technologies together. A large number of studies on the combination of WPT and MEC have been performed to date.

4 Conclusions

5G can offer an incredibly scalable and programmable device capable of integrating computing technologies with a wide variety of communication types (i.e. Human and Machine). The resulting network would be quicker and smarter, allowing less energy to be absorbed by higher magnitudes of linked objects than today. The industry therefore hopes that 5G would have a single Internet of Things reference structure. The MEC-enabled 5G network is an infrastructure for multi-level computing and connectivity. MEC-enabled 5G networks cooperatively integrate data transmission, data storage and data processing in a single network architecture compared to conventional mobile communication networks that focus solely on data transmission, significantly reducing end-to-end transmission latency, which is a key problem that 5G communication networks are struggling to cope with. While the MEC-enabled 5G network brings many network design advantages and has drawn considerable research interests, there are still some challenges that require more research investigation to realize the goals offered by MEC-enabled 5G networks, which we addressed in this chapter. To assign the system resources in accordance with the upper level application requirements, a unified management architecture is required. Furthermore, further theoretical work is also required to identify and model network utilization ability, how to convert network utility capacity and computing power, and how to use virtualization and network technology to achieve efficient allocation of resources, etc. Such problems all place great demands to deepen and refine the design of the 5G architecture allowed by MEC.

References

1. Zhang Y, Abbas N, Taherkordi A, Skeie T (2018) Mobile edge computing: a survey. IEEE Int Things J 5:450–465
2. Ha VN, Piran MJ, Le M, Le LB, Hwang W-J, Pham QV, Fang F, Ding Z (2020) A survey of multi-access edge computing in 5G and beyond: fundamentals, technology integration, and state-of-the-art. IEEE Communications Surveys & Tutorials
3. Chen S, Zhao J (2014) The requirements, challenges, and technologies for 5G of terrestrial mobile telecommunication. IEEE Communications Magazine 52(5):36–43
4. 5G America, white paper (2019). Available online: https://www.5gamericas.org/5g-at-the-edge/. (Accessed 5 July 2021)
5. Ericsson mobility report (2019). Available online: https://www.ericsson.com/49d1d9/assets/local/mobility-report/documents/2019/ericssonmobility-reportjune-2019.pdf. (Accessed 16 Sept 2019)
6. Nour B, Sharif K, Li F, Biswas S, Moungla H, Guizani M, Wang Y (2019) A survey of internet of things communication using ICN: a use case perspective, computer communications 142–143(2019):95–123. http://dx.doi.org/10.1016/j.comcom.2019.05.010, URL http://www.sciencedirect.com/science/article/pii/S0140366418309228
7. Nour B, Sharif K, Li F, Moungla H, Liu Y (2020) A unified hybrid information-centric naming scheme for IoT applications. Comput Commun 150(2020):103–114, http://dx.doi.org/10.1016/j.comcom.2019.11.020, URL http://www.sciencedirect.com/science/article/pii/S0140366419306553

8. Patel M, Naughton B, Chan C, Sprecher N, Abeta S, Neal A et al (2014) Mobile-Edge computing - introductory technical white paper. White Paper, Mobile-edge Computing (MEC) Industry Initiative
9. Hu YC, Patel M, Sabella D, Sprecher N, Young V (2015) Mobile edge computing a key technology towards 5G. ETSI White Paper 11
10. Ahmed A, Ahmed E (2016) A survey on mobile edge computing. In: 10th IEEE International Conference on Intelligent Systems and Control (ISCO), pp 1–8
11. Al-Shuwaili A, Simeone O (2017) Energy-efficient resource allocation for mobile edge computing-based augmented reality applications. IEEE Wire Commun Lett 6(3):398–401
12. Chen M, Challita U, Saad W, Yin C, Debbah M (2017) Machine learning for wireless networks with artificial intelligence: a tutorial on neural networks. [Online]. Available: http://arxiv.org/abs/1710.02913
13. Raman A, Sastry N, Sathiaseelan A, Chandaria J, Secker A (2017) Wi-stitch: content delivery in converged edge networks. In: Proceeding of the workshop Mobile Edge Communication ACM, pp 13–18
14. Sun Y, Chen Z, Tao M, Liu H (2018) Communication, computing and caching for mobile VR delivery: modeling and trade-off. In: Proceeding of the IEEE International Conference Communication, May 2018, pp 1–6
15. Yeow K, Gani A, Ahmad RW, Rodrigues J, Ko K (2017) Decentralized consensus for edge-centric internet of things: a review, taxonomy, and research issues. IEEE Access 6(6):1513–1524
16. Mao Y, You C, Zhang J, Huang K, Letaief K (2017) A survey on mobile edge computing: the communication perspective. IEEE Communications Surveys & Tutorials 19:2322–2358
17. Ahmed E, Rehmani MH (2016) Mobile edge computing: opportunities, solutions and challenges. Future Gene Comput Syst 70. https://doi.org/10.1016/j.future.2016.09.015
18. Cisco (2017) Visula Neworking index: global fixed and mobile internet traffic forecasts. Feb 2017. https://www.cisco.com/c/en/us/solutions/service-provider/visualnetworking-index-vni/index.html. [Accessed on Dec 2020]
19. Cisco (2019) Cisco visual networking index: global mobile data traffic forecast update. 20172022. Feb 2019. https://www.cisco.com/c/en/us/solutions/collateral/serviceprovider/visual-networking-index-vni/white-paper-c11-738429.pdf. [Accessed on Apr 2019]
20. Wei X et al (2017) MVR: an architecture for computation offloading in mobile edge computing. In: 2017 IEEE International Conference on Edge Computing (EDGE), Honolulu, HI, pp 232–235. https://doi.org/10.1109/IEEE.EDGE.2017.42
21. Sarukkai RR, Mendhekar A (2007) Method and apparatus for accessing targeted, personalized voice/audio web content through wireless devices. 27 Apr 2004, US Patent 6,728,731
22. Dev D, Patgiri R (2016) Dr. Hadoop: an infinite scalable metadata management for hadoop—how the baby elephant becomes immortal. Frontiers Inform Technol Electron Eng 17:15–31. https://doi.org/10.1631/FITEE.1500015
23. Silva R, Santos D, Corujo D, Aguiar RL, Figueiredo S, Parreira B (2019) Mobility-optimized dynamic content placement for fast vehicles in 5G networks. In: 2019 IEEE 30th annual international symposium on personal, indoor and mobile radio communications (PIMRC), Istanbul, Turkey, pp 1–7. https://doi.org/10.1109/PIMRC.2019.8904348
24. Kaur MJ, Maheshwari P (2016) Building smart cities applications using IoT and cloud-based architectures. In: 2016 international conference on industrial informatics and computer systems (CIICS 2016), Sharjah, UAE, pp 1–5. https://doi.org/10.1109/ICCSII.2016.7462433
25. Schulz P, Matthe M, Klessig H, Simsek M, Fettweis G, Ansari J, Ashraf SA, Almeroth B, Voigt J, Riedel I et al (2017) Latency critical IoT applications in 5G: perspective on the design of radio interface and network architecture. IEEE Commun Mag 55(2):70–78
26. Porambage P, Okwuibe J, Liyanage M, Ylianttila M, Taleb T (2018) Survey on multi-access edge computing for internet of things realization. IEEE Commun Surv Tutorials 20(4):2961–2991, Fourth quarter 2018. https://doi.org/10.1109/COMST.2018.2849509
27. Beck MT, Werner M, Feld S, Schimper T (2014) Mobile edge computing: a taxonomy

28. Sabella D, Vaillant A, Kuure P, Rauschenbach U, Giust F (2016) Mobile edge computing architecture: the role of MEC in the internet of things. Consum Electron Mag 5(4):84–91
29. Mach P, Becvar Z (2017) Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun Surv Tutorials 19(3):1628–1656
30. Taleb T, Samdanis K, Mada B, Flinck H, Dutta S, Sabella D (2017) On multi-access edge computing: a survey of the emerging 5G network edge architecture and orchestration. Commun Surv Tutorials 19(3):1657–1681
31. Baktir AC, Ozgovde A, Ersoy C (2017) How can edge computing benefit from software-defined networking: a survey, use cases and future directions. Commun Surv Tutorials 19(4):2359–2391
32. Afolabi I, Taleb T, Samdanis K, Ksentini A, Flinck H (2018) Network slicing and softwarization: a survey on principles. Enable Technol Sol. In IEEE Communications Surveys & Tutorials, vol 20, no 3, pp 2429–2453, thirdquarter 2018. https://doi.org/10.1109/COMST.2018.2815638
33. Roman R et al (2018) Mobile edge computing, Fog et al.: a survey and analysis of security threats and challenges. Future Gener Comput Syst 78:680–698
34. Mao Y, You C, Zhang J, Huang K, Letaief KB (2017) A survey on mobile edge computing: the communication perspective. Commun Surv Tutorials 19(4):2322–2358
35. Wang S, Zhang X, Zhang Y, Wang L, Yang J, Wang W (2017) A survey on mobile edge networks: convergence of computing, caching and communications. Access 5:6757–6779
36. Abbas N, Zhang Y, Taherkordi A, Skeie T (2018) Mobile edge computing: a survey. IEEE Internet Things J 5(1):450–465
37. Ai Y, Peng M, Zhang K (2017) Edge cloud computing technologies for internet of things: a primer. Digital Commun Network
38. Subramaniam P, Kaur MJ (2019) Review of security in mobile edge computing with deep learning. In: 2019 Advances in Science and Engineering Technology International Conferences (ASET), pp 1–5. https://doi.org/10.1109/ICASET.2019.8714349
39. Shahzadi S, Iqbal M, Dagiuklas T et al (2017) Multi-access edge computing: open issues, challenges and future perspectives. J Cloud Comp 6:30. https://doi.org/10.1186/s13677-017-0097-9
40. Garcia Lopez P, Montresor A, Epema D, Datta A, Higashino T, Iamnitchi A, Barcellos M, Felber P, Riviere E (2015) Edge-centric computing: vision and challenges. SIGCOMM Comput Commun Rev 45(5):37–42
41. Ding Z, Fan P, Poor HV (2016) Impact of device pairing on 5G nonorthogonal multiple-access downlink transmissions. IEEE Trans Veh Technol 65(8):6010–6023
42. Vaezi M, Schober R, Ding Z, Poor HV (2019) Non-orthogonal multiple access: common myths and critical questions. IEEE Wireless Commun 26(5):174–180
43. Ding Z, Schober R, Fan P, Poor HV (2019) Simple semi-grantfree transmission strategies assisted by non-orthogonal multiple access. IEEE Trans Commun 67(6):4464–4478
44. Ding Z, Fan P, Karagiannidis GK, Schober R, Poor HV (2018) NOMA assisted wireless caching: Strategies and performance analysis. IEEE Trans Commun 66(10):4854–4876
45. Mostafa AE, Zhou Y, Wong VWS (2019) Connection density maximization of narrowband IoT systems with NOMA. IEEE Trans Wirel Commun 18(10):4708–4722
46. Zhai D, Zhang R, Cai L, Yu FR (2019) Delay minimization for massive internet of things with non-orthogonal multiple access. IEEE J Sel Topics Signal Process 13(3):553–566
47. Kiani A, Ansari N (2018) Edge computing aware NOMA for 5G networks. IEEE Internet Things J 5(2):1299–1306
48. Qian LP, Feng A, Huang Y, Wu Y, Ji B, Shi Z (2019) Optimal SIC ordering and computation resource allocation in MEC-aware NOMA NB-IoT networks. IEEE Internet Things J 6(2):2806–2816
49. Wu Y, Ni K, Zhang C, Qian LP, Tsang DHK (2018) NOMA Assisted multi-access mobile edge computing: a joint optimization of computation offloading and time allocation. IEEE Trans Veh Technol 67(12):12244–12258
50. Diao X, Zheng J, Wu Y, Cai Y (2019) Joint computing resource, power, and channel allocations for D2D-assisted and NOMA-based mobile edge computing. IEEE Access 7:9243–9257

51. Peng M, Sun Y, Li X, Mao Z, Wang C (2016) Recent advances in cloud radio access networks: System architectures, key techniques, and open issues. IEEE Commun Surv Tut 18(3):2282–2308
52. Zhang H, Qiang Z, Du X (2015) Toward vehicle-assisted cloud computing for smartphones. IEEE Trans Veh Technol 64(12):5610–5618
53. Zhang J, Xu Y, Xia W, Xu Y, Cai S, Zhu H (2019) Resource allocation scheme of downlink C-RAN with edge caching and fronthaul compression. IEEE Access 7(1):118669–118678
54. Hou X, Yong L, Min C, Di W, Jin D, Sheng C (2016) Vehicular fog computing: a viewpoint of vehicles as the infrastructures. IEEE Trans Veh Technol 65(6):3860–3873
55. Mach P, Becvar Z (2017) Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun Surv Tut 19(3):1628–1656
56. Li S, Ning Z, Lin S, Kong L, Katangur A, Khan MK, Ni M, Gang Z (2018) Joint admission control and resource allocation in edge computing for internet of things. IEEE Netw 32(1):72–79
57. Barbarossa S, Sardellitti S, Lorenzo PD (2014) Communicating while computing: distributed mobile cloud computing over 5G heterogeneous networks. IEEE Signal Proc Mag 31(6):45–55
58. Tang J, Quek TQS, Chang T-H, Shim B (2019) Systematic resource allocation in cloud RAN with caching as a service under two timescales. IEEE Trans Commun 67(11):7755–7770
59. Chen L, Zhou S, Xu J (2015) Decentralized computation offloading game for mobile cloud computing. IEEE Trans Parallel Distrib Syst 16(8):974–983
60. Wu Q, Zeng Y, Zhang R (2018) Joint trajectory and communication design for multi-UAV enabled wireless networks. IEEE Trans Wirel Commun 17(3):2109–2121
61. Pan C, Ren H, Deng Y, Elkashlan M, Nallanathan A (2019) Joint blocklength and location optimization for URLLC-enabled UAV relay systems. IEEE Commun Lett 23(3):498–501
62. Li A, Wu Q, Zhang R (2019) UAV-enabled cooperative jamming for improving secrecy of ground wiretap channel. IEEE Wirel Commun Lett 8(1):181–184
63. Lee H, Eom S, Park J, Lee I (2018) UAV-aided secure communications with cooperative jamming. IEEE Trans Veh Technol 67(10):9385–9392
64. Zhou X, Wu Q, Yan S, Shu F, Li J (2019) UAV-Enabled secure communications: joint trajectory and transmit power optimization. IEEE Trans Veh Technol 68(4):4069–4073
65. Zhong C, Yao J, Xu J (2019) Secure UAV communication with cooperative jamming and trajectory control. IEEE Commun Lett 23(2):286–289
66. Choi KW et al (2017) Wireless-powered sensor networks: how to realize. IEEE Trans Wireless Commun 16(1):221–234
67. Huang J, Zhou Y, Ning Z, Gharavi H (2019) wireless power transfer and energy harvesting: current status and future prospects. IEEE Wirel Commun 26(4):163–169. https://doi.org/10.1109/MWC.2019.1800378
68. Ramezani P, Jamalipour A (2017) Toward the evolution of wireless powered communication networks for the future internet of things. IEEE Net 31(6):62–69
69. Choi KW, Ginting L, Aziz AA, Setiawan D, Park JH, Hwang SI, Kang DS, Chung MY, Kim DI (2019) Toward realization of long-range wireless-powered sensor networks. IEEE Wireless Commun 26(4):184–192
70. Zhou Z et al (2019) Energy-efficient resource allocation for energy harvesting-based cognitive machine-to-machine communications. In IEEE Transactions on Cognitive Communications and Networking, vol 5, no 3, pp 595–607. https://doi.org/10.1109/TCCN.2019.2925025

Methodology to Ensure the Continuity of the Information Systems Service, Based on the Monitoring of Electrical Energy, Using IoT Technology

Wilver Auccahuasi, Kitty Urbano, Edward Flores, Luis Romero, Monica Diaz, Edwin Felix, Nicanor Benites, Fernando Sernaque, Denny Lovera, Orlando Pacheco, and Mario Ruiz

Abstract In health centers, hospital services are a useful source of information that cannot be interrupted, as they provide necessary clinical information such as results of clinical examinations, results of radiological examinations, search results for available drugs, among other necessary information. In the absence of commercial electrical energy, mechanisms should be considered to ensure the continuity of information services through the use of battery banks, this methodology presents a technique based on IoT techniques, through which the consumption of the package is monitored. of batteries found in the data centers of hospital establishments, to be able to analyze the level of charge and the state of the batteries at all times when the

W. Auccahuasi (✉) K. Urbano
Universidad Privada del Norte, Lima, Perú
e-mail: wilver.auccahuasi@upn.edu.pe

K. Urbano
e-mail: kitty.urbano@upn.edu.pe

E. Flores L. Romero D. Lovera O.·Pacheco
Universidad Nacional Federico Villarreal, Lima, Perú
e-mail: eflores@unfv.edu.pe

L. Romero
e-mail: lromero@unfv.edu.pe

D. Lovera
e-mail: dlovera@unfv.edu.pe

O. Pacheco
e-mail: opacheco@unfv.edu.pe

M. Diaz
Universidad Norbert Wiener, Lima, Perú
e-mail: monica.diaz@uwiener.edu.pe

E. Felix M. Ruiz
Universidad Nacional de Moquegua, Moquegua, Perú
e-mail: efelixp@unam.edu.pe

M. Ruiz
e-mail: mruiz@unam.edu.pe

R. Tiwari et al. (eds.), *Energy Conservation Solutions for Fog-Edge Computing Paradigms*, Lecture Notes on Data Engineering and Communications Technologies 74,
https://doi.org/10.1007/978-981-16-3448-2_14

commercial electrical power is active, when the commercial power is cut, the battery bank power, so your tracking on the consumption of the battery Charging is of vital importance to be able to activate the alternative and energy mechanisms such as electric power generating motors, but these motors have an ignition and stabilization time, it is at this time that it is necessary to monitor the use and discharge of energy from the power bank batteries found in data centers, the mechanism to be used for the connection through IoT is based on an ATECC608A development device, which has an ATWINC1510 Wi-Fi connection with an interface developed in LabView, The results demonstrate the use and practicality as most power outages occur at night so data center workers are not at their workplace and remote monitoring of battery banks is very useful in these situations based on IoT technology.

Keywords IoT · Monitoring · Temperature · Web server · Control · Information systems

1 Introduction

Information systems are of utmost importance in order to continue with the procedures that ensure business continuity, in this sense it is necessary to ensure the supply of electricity, to meet these objectives. Most companies configure their computer services in a server room better known as "Data Center", in its development many technologies are presented that are related to ensuring its operation. Carrying out a review of the literature, we present the following works with influence on the use of technology for the good performance of data centers.

The data centers present solutions and configurations of different types, such as the forms of energy supply, which divides the power delivery into 3 categories, centralized energy to the Architecture where it conventionally comprises a single power in the conversion stage, the Distributed energy architecture which is the most suitable for the delivery of energy in data centers and the Intermediate Bus architecture where it involves multiple buses and energy conversion levels higher than the previous ones, the 3 have been compared in voltage levels, in the work of the converter and the topologies used, carrying out an analysis on the architecture, for which the architecture has been detailed on a mathematical design, with an experimental validation of the data center, taking into account the systems where many servers are coupled, connected to a main server [1–6].

Evaluating the energy consumption of data center servers is very important, due to the problem of the heat it generates and the need to be able to optimize thermal

N. Benites
Univeraidad Nacional Mayor de San Marcos, Lima, Perú
e-mail: nbenites@unmsm.edu.pe

F. Sernaque
Universidad César Vallejo, Lima, Perú
e-mail: fsernaque@ucv.edu.pe

management, which were evaluated by data from the Standard Performance Evaluation Corporation (SPEC), taking into account that the more consumption, the more energy, with the advancement of technology these models must be renewed, together with the refrigeration equipment that helps in energy efficiency, where the following conclusion is reached, carrying out an analysis of the consumption of the servers, to be able to classify them according to the consumption [7–15].

The evaluation of the temperature is fundamental, where the airflow is evaluated, which is very important for the operation of data centers, with the airflow that moves between its thermal environment, which must be between 30 and 45 cm from the floor, for which a previous review was made on airflow factors, numerical study, airflow performance metrics and thermal optimization, with multiple characteristics can be classified in room-level environments, rack level and server level, factors such as design, raised floor, ceiling height, and tiles should be taken into account for the thermal environment. The level of the racks are evaluated according to the porosity of the door, the airflow rate and temperature, the number of servers, the server layout and the energy density, in the servers the speed of the airflow and the server fan, converting into numerical values to make calculations, using data center thermal behavior prediction methods, so we want to emphasize the importance of data center airflow serving as references for future airflow designs and energy efficiency in data centers [16].

Cooling devices for good temperature and humidity control in order to maintain stable conditions for the data center; for which the American Society of Heating, Refrigeration and Air Conditioning Engineers (ASHRAE) has developed thermal guidelines for these types of environments with classifications of classes A1-A4 according to their temperature level between 18 and 27 degrees Celsius and humidity in average of 60% (relative humidity), since these sites have greater sensitivity than other buildings, for this reason these places have refrigeration systems which work 24 h a day, 365 days a week, for which It must have a team that mismatches between supply and demand, that is, these teams must adapt the variations between the outside temperature and requirements to maintain a balance, this process has 5 mismatches between supply and demand, less efficiency in day than at night, free cooling is used only at night or in cold season, Peak valley usage time rate, Higher heat load from center d e data during the day and emergency cooling which is seen as cornerstones of maintaining data center reliability [6, 7, 9, 16–18, 18–22].

Data centers have become an indispensable part for social development, and today there is a great development of communication, data calculation, size and quantity growing exponentially, so energy consumption is also growing exponentially. Exponentially, for every square meter, energy consumption has been multiplied by 100, improving efficiency in the use of energy in these centers, so different methods are proposed and designed, there are still no statistical studies on thermal management and evaluation of data centers and cooling systems there are studies where the state of the art on the aforementioned topics is analyzed, containing energy conservation of energy and optimization methods, finally having as a conclusion as the effect of airflow on the floor, it will distribute thermally according to space and racks, in order to optimize in a data center to hot and cold aisle configuration. Thermal management focused on the floor and room space, with the main factor being the organization of the airflow under the raised floor, physical and design parameters under the floor should also be considered first, and improvements to the mixing of air, negative pressure and hot spots in the room space are the effective measures, in order to optimize has been divided into air cooling and cooling techniques for air and liquid cooling, in order to eliminate the hot places of the servers, with respect to the cooling liquid it is considered the most suitable for reducing the temperature of place for the elimination of heat effectively [22–24].

In the architecture of IoT technologies, we can indicate the use of multiple sensors configured in the form of aggregation and management nodes, the architecture has a beginning with the sensor node which is the basis of a network of nodes which is used to collect information, process it and send it to other higher nodes, the mesh topology is a multi-hop structure, these nodes can communicate with each other getting the path according to the protocols, algorithms and data forwarding, this type of topology is recommended For large scale networks, its main use being places that are outside the monitoring area, considering the network between machines which is called M2M, which are combined with various types of communication technologies between machines, which share information either machine-to-machine, man–machine, when talking about the M2M network, it is machine-to-machine communication without human intervention, the M2M network architecture cu enta with M2M domain is a reliable channel is based on the Internet and mobiles, the domain is a reliable channel to transmit the data from the M2M domain to the application domain, they maintain the transmission and manage the protocols and domain of the application, it is used as back-end services platform, to store and load by the M2M domain in real time for the M2M remote monitoring and management applications, according to each product request it can be analyzed, processed, controlled and assisted in decision making, it is considered as important branch of IoT [25, 26].

The area of health care for people is one of the areas where IoT technology is being exploited with great intensity, achieving interesting results, managing to improve information services, as well as patient care, making the most of connectivity and availability of medical devices with communication interfaces managing to integrate solutions that improve patient care [27].

In the management of the generator and energy converter equipment, we can mention the substations, which perform these tasks constantly, in order to be controlled remotely, solutions based on PLC in SCADA configurations are normally used, making these solutions high. With the use of IoT technology, low-cost solutions can be implemented, we found works where they use IoT technology in monitoring the output current and those produced by each transformer [28]. We also find applications where IoT technology is used in transformer monitoring, obtaining information on operating parameters such as working voltage values, as well as working environmental conditions, such as temperature, humidity and light levels. [29].

Making a summary of the aforementioned references, we can indicate that the main concern on the part of the researchers is to be able to maintain the temperature of the data centers, reasons for which multiple solutions are developed based on the management of refrigeration, with The purpose of controlling the temperature, starting from a simple logic, the higher the consumption of computer services, the servers require more power, this causes a higher current consumption, which triggers a generation of higher working temperature, then we can indicate that the higher the consumption, the higher the temperature and the higher the temperature requires more cooling in terms of energy, the higher the consumption of computer services requires higher energy consumption, therefore being able to control these parameters is of vital importance to be able to control and optimize energy consumption. In the proposal presented, a mechanism is described to be able to monitor electrical parameters of energy consumption, as well as environmental parameters that are necessary for the proper functioning and assurance of the continuity of computer services.

2 Materials and Methods

The proposed methodology consists of maintaining the information services, at all times and in the face of any unconventional situation, such as the absence of commercial power, which is the moment when the energy backup systems come into operation, or better known as UPS. While the energy continuity systems begin to work, these systems based on substations work with diesel engines, which produce a level of pollution, the greater the amount of load or energy that is needed, the engine will emit more polluting gases, For this reason, it is important to monitor the operating parameters of the data centers or "Data Center", such as the temperature of both the servers and the systems that provide cold air in order to keep the Data Centers at the appropriate temperature.

For the development of the proposed methodology, it is necessary to have several components from the analysis of the literature, through the choice of IoT devices to the configuration of the solution and the interpretation of the results.

In order to ensure the continuity of services in the information systems, monitoring the electrical parameters, Fig. 1 shows the block diagram of the procedures to follow in order to explain the proposed methodology in detail.

2.1 Literature Review

In the design of the methodology, one of the important tasks is the study of the "Study of Art", which provides us with necessary and updated information on the mechanisms currently used to evaluate the continuity of information systems. They took 35 publications, finding as management indicators that computer services are organized in a physical space called Data Center, which is subject to a series of controls necessary for its correct operation, these controls are related to the temperature of the servers, temperature of the refrigeration systems, the forms of configuration with the alternate units that provide electrical energy, among others. The literature review allowed an optimization in the design of the proposed methodology.

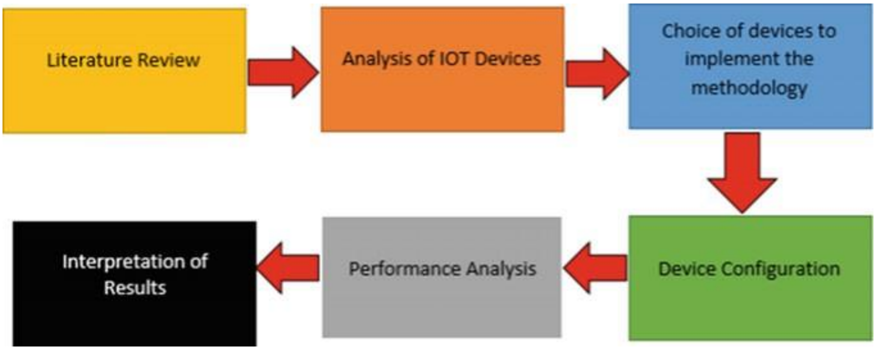


Fig. 1 Block diagram of the proposed methodology

2.2 Analysis of IoT Devices

In the electronic components and devices market, there are a variety of devices dedicated to providing IoT-based solutions, each with a different function and configuration, making it possible to solve multiple problems. In our particular case, in the search for these devices, our need and subsequent requirement are characterized in being able to continuously capture a series of physical and electrical parameters in order to be able to analyze the information and be able to make better decisions in the administration of the energy in our Data Center, in order to safeguard the continuity of the Information Systems.

2.3 Choice of Devices to Implement the Methodology

In choosing the electronic device, which will be in charge of collecting the information and can be uploaded to a web server for later exploitation; It was considered 3 main characteristics that is intelligent, safe and connected, for which the chosen card is the PIC-IoT of the Microchip brand that by its design has the following specifications:

- Smart, this feature is adopted because the device has the PIC24FJ128GA705.
- Sure, this feature is adopted because the device has the ATECC608A chip
- Connected, this feature is adopted because the device has the WINC1510 Wi-Fi chip

In addition to these three characteristics, it can be indicated that the IoT device has low consumption, is small compared to other devices that perform the same function, has a serial communication interface, and has the facility to be recognized by the operating system as a unit storage, it has several libraries and development environments to perform configuration and programming, such as MPLAB, X IDE among others (Fig. 2).

The control mechanism is another of the strengths of the methodology, to carry out this task it is necessary to be able to have a device that can be controlled remotely and effectively, for this task the best known energy control devices were chosen like Smarte Socket, which are made up of an intelligent "relay" that is activated from a Wi-Fi connection, this Smart Socket can be configured from an IP address, through various IoT applications, that is why if it has an IP fixed can be controlled remotely.

The work carried out by this device is to provide power to the servers, in such a way that they can function, according to the priority of the services that they want to continue, to keep them on or off, in order to manage the best way available energy, both from the substations and from the "UPS" battery Banks (Fig. 3).

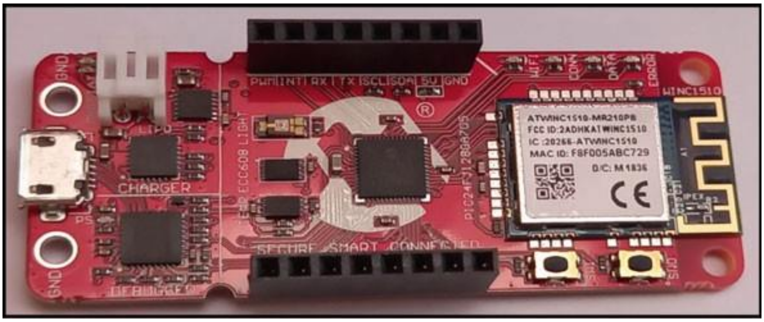


Fig. 2 PIC-IoT microchip development board, with ATWINC1510 Wi-Fi module

Fig. 3 Smart socket, for server power control

2.4 Device Configuration

The configuration of the devices is important, to comply with all the characteristics of the methodology, in Fig. 4 the diagram is presented, where all the components are presented and distributed in an orderly and logical way to perform each of the proposed tasks. It should be taken into account that each of the connections has an

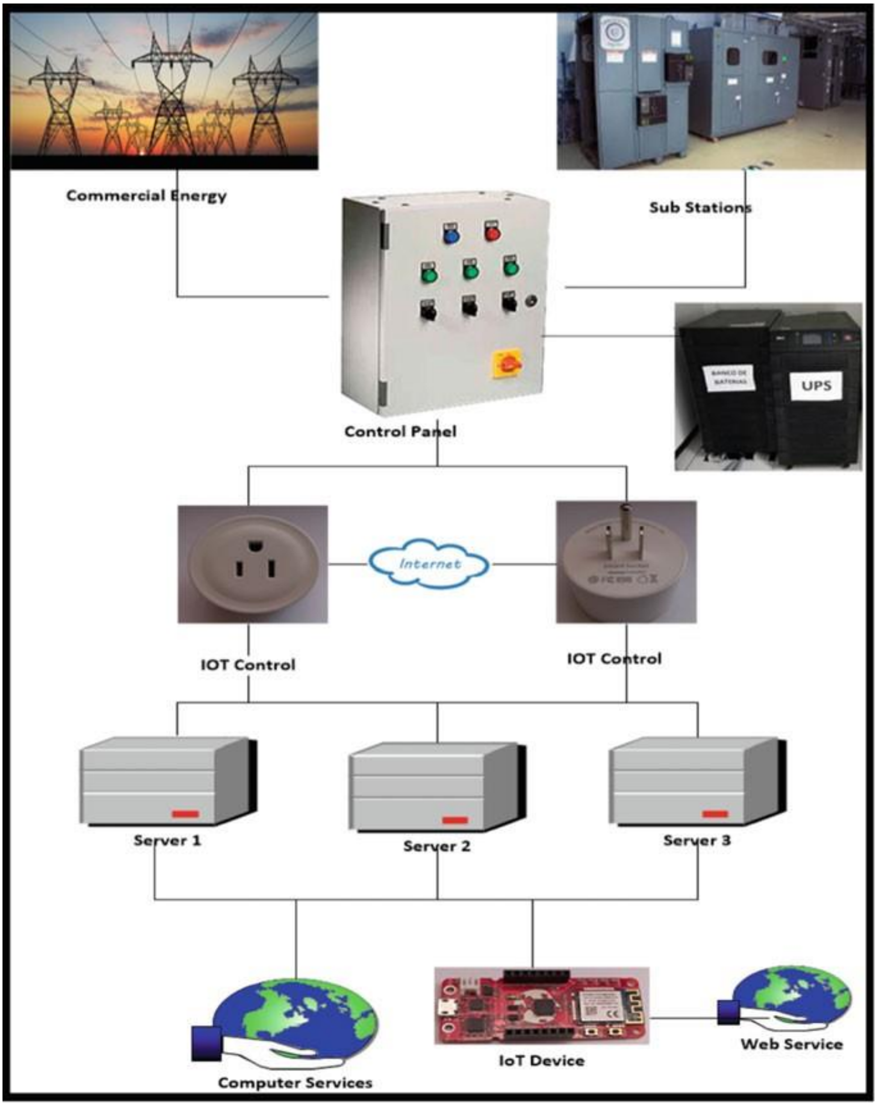


Fig. 4 Diagram of the configuration of the methodological proposal

objective and this will be necessary for the next connection, the components that make up the methodology are made up of: the commercial power line, the substation that provides energy by means of a diesel engine, control panel, IoT devices, servers and IT services.

The configuration allows to simulate a real scenario of a Data Center and all the components necessary for its operation, in conventional mode and in the event of any emergency situation, caused by a fluid cut in the commercial line, where the battery bank comes to work and the substation, likewise the control panel is made up of the Smart Socket who is in charge of providing power to the servers and blocking the power when they do not require it in an emergency situation, a representation of the servers is also presented where each one provides a service, these servers are connected to the signal capture unit based on IoT technology, where it has the function of collecting the operating parameters of the Data Center and sends them to a web server to be able to store the data for later exploitation.

In Fig. 4, the detail of the configuration of the various components is presented, then we describe each of the connections:

In the first segment, we have the connection of the commercial line and the substation to the control panel, this is in charge of being able to select between the entrance of the commercial line and the line that provides the substation, the commercial line is The normal line provided by the electrical service companies, the line that comes from the substation using diesel engines, this alternate line causes fuel consumption according to the load capacity that is required to be provided, in the event of a possible power outage the commercial line and while the substation is turned on until its output is stable, a power supply is necessary that allows continuity of the computer services, it is at this time that the battery banks better known as UPS come to work, they provide power to the servers to ensure its operation.

This is the most important moment of the methodology, and it is to be able to select the servers that have to remain online and which of them has to be turned off, because the UPS has a limited operating autonomy and in most cases it is used To ensure the correct shutdown of the servers instead of forcibly shutting down, the Smart Socket works in this task, which are controlled through a Wi-Fi connection, in this way it is possible to select which of the servers is powered by the UPS line and which are turned off, this connection can be considered as an intelligent control if the choice is made automatically.

In a second segment, we have the servers, they provide the different computer services ensuring the continuity of the information flow. The servers are located in the data centers and work 24 h a day, 365 days a year, this characteristic makes it necessary to monitor the correct functioning of the servers. One of the necessary parameters to be able to monitor is the operating temperature of the servers, the proposed methodology is dedicated to being able to monitor the temperature in the vicinity of the servers, the temperature in the air that circulates in the Data Center room and the temperature provided by the cooling system because, this mechanism to monitor the temperature in the 3 evaluation points, allows controlling the cooling effect, in order to be able to analyze the energy consumption of the servers based on the heat it produces, starting. The greater the work of the servers, the greater the heat generated by the work done by the electronic components. These data are collected by Microchip's PIC-IoT device, which collects the temperature in the 3 points and stores them in a web service, in order to be exploited for better decision making, in an emergency situation. Product of a lack of energy in the commercial line.

2.5 Performance Analysis

The performance of the analysis of the methodology is dedicated to being able to establish a workflow and to be able to react to any emergency situation, we understand emergency as a situation where it is necessary to ensure the continuity of computer services, when there has been a cut in the commercial line, requiring analysis of which servers need to continue working and which need to be shut down.

The flow diagram that is represented in Fig. 5, consists of a series of steps that must be carried out sequentially and logically in order to ensure the continuity of computer services, in the presence of a power outage of the commercial line.

The procedure begins with the failure of commercial power, causing the power cut provided by the electric power companies, this power cut causes a decision on which servers and information services should continue to function and which ones need to be turned off in order to meet the fundamental objective that is to ensure the continuity of computer services.

In this analysis, the emergency begins with the commercial power cut, it is at this moment where the control unit begins to work, starting two tasks, the first is by starting to work the battery bank and sending the ignition of the substation, this ignition has a considerable time between the starting of the motor and the stabilization of the current, this time will depend on the brand, model, and the maintenance carried out to the substation, normally this time ranges from 2 to 5 min, this time is very important because to maintain computer services it is necessary that the battery side can supply power to the Data Center, in the Data Center there are many servers, each with a specific functionality, mail server, domain server, application server, database server, each of these servers has a priority of operation according to the time of the emergency and the services provided on necessary deliver at the indicated time.

In the 2 to 5 min it takes for the substation to come online, the battery bank is providing power to all the servers in the data center. Depending on the time the emergency occurs, there are two cases, the first is that the emergency occurs during working hours where the staff can work and perform analysis operations and manual shutdown of the servers, if the emergency occurs in hours where there is no staff to operate the servers, the control must be carried out remotely, this control is carried out by analyzing the information that is available thanks to the IoT modules, with this analysis it can be decided remotely which of the servers work can continue and which ones can be turned off, while the emergency is regularized, through the return of commercial power or the stabilization of power from the substation, this decision is important and will also depend on the capacity of the battery bank, analyzed for capacity, run time, remaining run time.

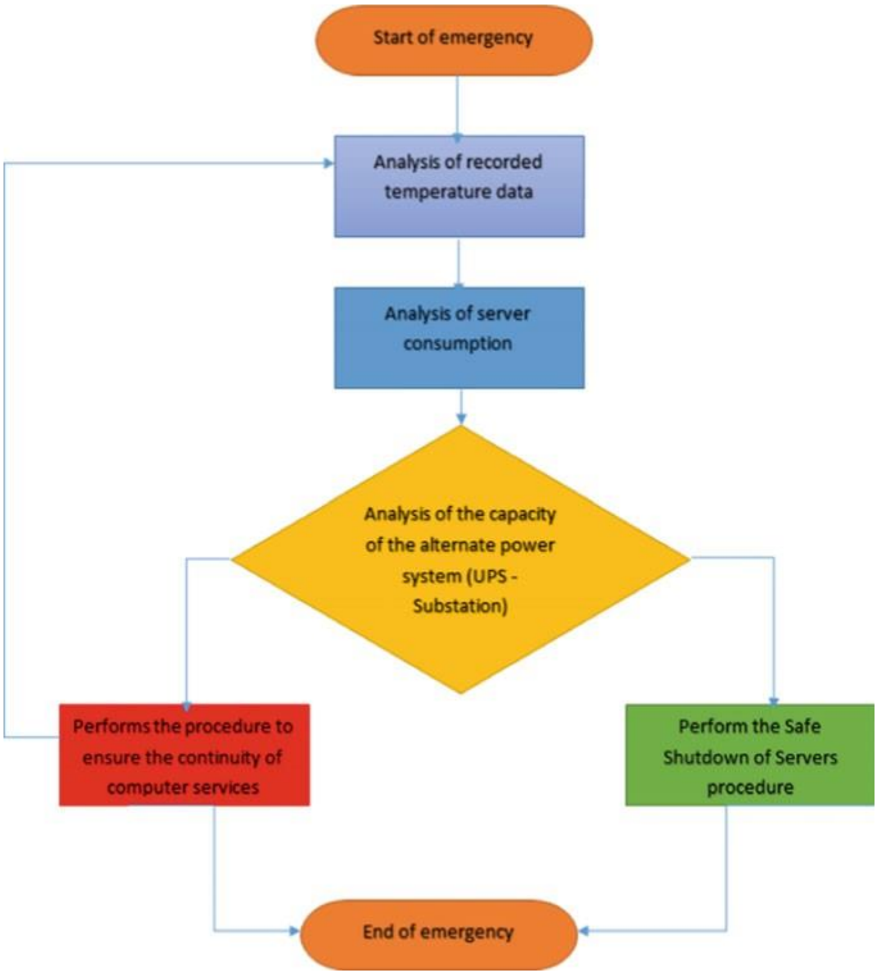


Fig. 5 Methodology workflow

After analyzing the priorities of the servers, the servers that are with a lower operating priority are turned off, when the power of the substation comes into operation, these servers can be turned on again and the services provided by the Data Centers can continue. When a substation is not available, all servers are shut down until the commercial line is replaced, because the battery banks provide an average of 20 to 30 min of power, the time necessary for the correct shutdown of the servers, It should be mentioned that a forced shutdown of the servers can cause damage to the Software that it has installed, causing mostly that when it is turned on again it does not work correctly due to the wrong shutdown of the operating system, it is essential to indicate that the servers are always in operation and working 24 h a day, 365 days a year.

2.6 Interpretation of Results

The methodology proposes that the data collected by the IoT unit are stored in a database and accessible through a web service, these data can be presented in various formats, such as Excel, notepad, among others, it will depend mainly on skills computational staff, it is important to indicate that the decision to choose which of the servers have priority of operation must be made periodically and that it is constantly updated, in such a way that at the time the emergency is presented, the list of servers with their priority level.

In being able to analyze the data obtained by the IoT unit, it will allow to organize the servers according to their operating priorities in a certain task, they can be analyzed following several criteria, one of them is the analysis by operating hours, it can be analyzed and grouped the servers according to the work time and make a priority designation, for example, we can indicate that in work hours, the domain server where the workers are registered has a high priority, the application server where the computer systems are running like this as the database server has a high priority in non-business hours.

3 Results

The results that are presented are related to being able to demonstrate that the proposed methodology presents many solutions and various alternatives, through the exploitation of the recorded data that can serve for better decision making, to know the performance of the Data Center. To know the energy consumption, the mode of operation, the thermal operating conditions, which is a very important point in the operation of the Data Center, among other functionalities that can be exploited with the data obtained, in Fig. 6, it presents a description by means of a block diagram of a typical solution and how both the Data Center modules and the IoT component modules are integrated, and in order to establish this connection, it is important to connect the Data Center control unit and the Acquisition Unit of the IoT module.

This connection is very important and is the basis of the proposed methodology.

Figure 6 presents a block diagram of the conventional modules of the data centers that provide the information systems, these configurations can be found in the different data centers, and as a result it is proposed to use a type of characteristic interface made up of the RS232 protocol, to be able to transmit the information from the data center to the IoT unit.

In the IoT module, we find as the main unit, the acquisition unit, who is in charge of collecting the information from the control unit of the information systems module, and additionally, the IoT module comprises two more units, the unit of components of control and the display unit, who are in charge of being able to control the operation of the Smart Socket, as well as the display of energy consumption and the different

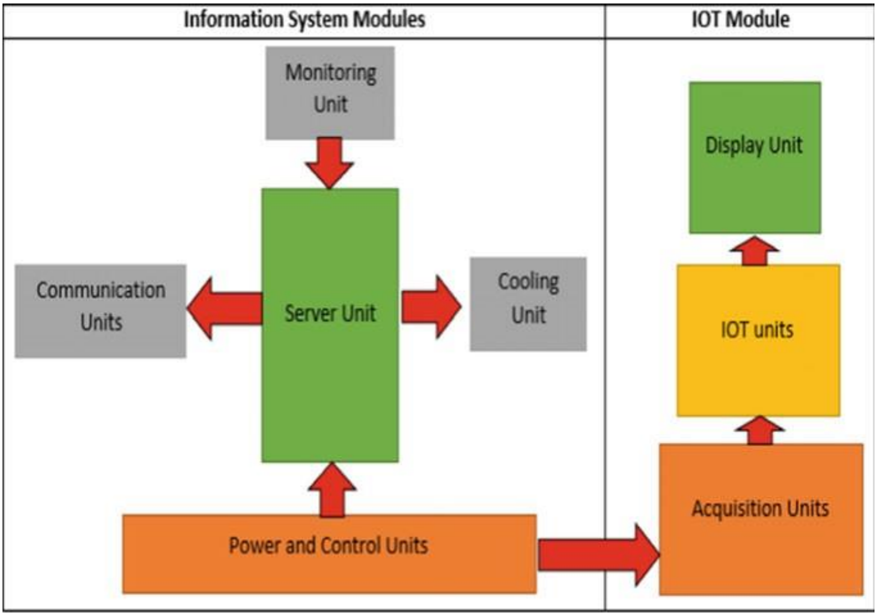


Fig. 6 Block diagram of the model connection in the implementation of the methodology

temperature measurements present in the server room. Below we describe each of the blocks in Fig. 6.

3.1 Server Unit

The server unit is the fundamental axis in the operation of information systems, they are in charge of being able to provide the different computer services that today, we are used to using and they are largely responsible for making purchases online, book some type of tickets, view e-mails, carry out banking transactions, any action we carry out remotely, from a computer equipment to a mobile device such as cell phones, pass through the servers.

To ensure the continuity of these services, it is important that the servers are always working with the appropriate temperature, this temperature factor is very important because if it rises either for different reasons, it puts its operation at risk from the physical part, that would cause a physical damage as a result of a malfunction of electronic components, as theoretical information, among the operating characteristics of electronic equipment and components, the most critical are, operating temperature, voltage and working current, if they begin to several out of their working range, there is a risk that these will fail, causing some component of the server to be defective, among the most sensitive components of the servers that can suffer from a sudden variation in temperature, are the memories, solid drives, mainboard and power supply.

The servers according to the brand, model, generation, work in different ways, having as the main characteristic in common, the working temperature, therefore it is of the utmost importance that the environment designated to locate the servers has air conditioning throughout. Moment, while the servers are running, which are 24ha day, 365 days a year. The working temperature is a concern, because the server equipment is stacked vertically inside compartments called RACKS, the servers are organized depending on their size, causing the temperature in the rack to be considerably high compared to the temperature of a single server.

In Fig. 7, the front view of a rack is presented, with the servers inside, stacked in a vertical order, it can be seen that 4 servers in the lower part and their corresponding disk arrangements in the upper part, is also incorporated The network connection units so that they can connect to the network, each of the servers and the disk arrays,

Fig. 7 Front view of a rack with servers installed

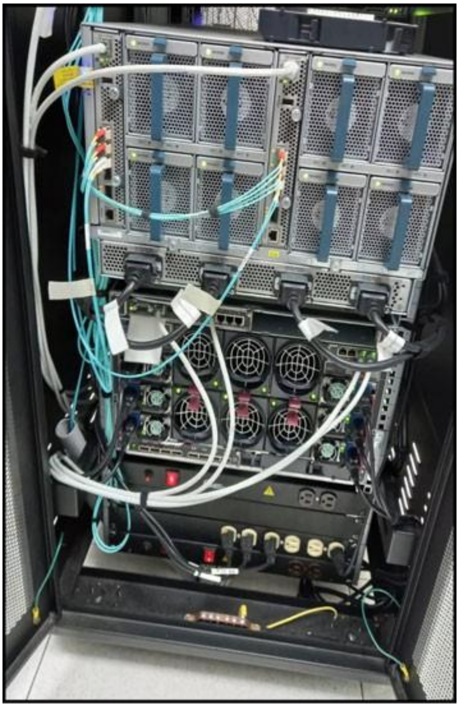


Fig. 8 Rear view of a rack with servers installed

produce heat for the work done, by which each one contributes heat to the temperature of the rack and therefore to the temperature from the server room.

In Fig. 8, you can see the rear view of the rack, where you can see the connections of the servers to the power source, connections of the disk arrays, it can be seen that each of these equipment has a ventilation unit, which allows the exchange of air, sending fresh air and removing hot air, this operating mode is very important, because if the air conditioning stops providing fresh air, the temperature of the medium increases, consequently the ventilation unit of the equipment that is in the rack, begins to provide warm air into the servers, instead of fresh air, causing the effect of heating the server temperature even more, if this situation happens to each server and disk array, the temperature begins to rise considerably, that is why data centers are always air-conditioned in order to control the temperature, it is converted erte in a critical operating factor.

3.2 Power and Control Units

The power unit and control panel is in charge of distributing the energy from the commercial line, the substation line and the battery bank line, as can be seen in Fig. 9. The control panel also has a communication unit, allowing data center operation data

Fig. 9 Control panel view

to be sent to other external mechanisms, thanks to the RS232 protocol, which has, as well as the DB9 connector through which it can be connected to a serial output, such as can be seen in Fig. 10.

In Fig. 10, you can see the circuit of the communications unit, where the information from the data center is concentrated, such as the different temperatures, the voltages from the different lines, the working current as well as the power total consumption, all this data can be exported through a frame to the serial output through the DB9 connector.

This unit works in tune with the IoT module reception unit, which also has a serial input through the RS232 protocol, which allows data reception and through the MQTT protocol they are exported and stored for future exploitation, analysis and capture of decisions, in Fig. 11, you can see the block diagram between these two devices.

Fig. 10 View of the communication unit from the control panel

Fig. 11 Communication protocol between the control panel and the IoT module

Thanks to the use of the RS232 protocol, it is possible to connect these two units and share information, the task in this communication configuration is to put the control panel circuit in transmission mode and in reception mode, on the IoT unit chip.

3.3 Monitoring Units

In a data center, the monitoring unit is in charge of verifying that the services provided by the servers are running without problems, so a monitor connected to the servers is required to verify their operation, between tasks of verification, which is the status of the operating system, the status of the database, service updates, storage availability, among others.

3.4 Communication Units

The communication unit is in charge of providing connectivity between servers and workstations within a network, and for this reason, the communications equipment composed of concentrators and routers are arranged in a rack, very similar to servers, with the purpose of being able to organize the route of the network cables, these equipments also consume energy and therefore generate heat for the same operation. Making an analysis at the temperature of the data center, we must indicate that the heat generated by the communication unit is added to the heat generated by the server unit, and it is a function of the air conditioning system, to be able to control this heat (Fig. 12).

3.5 Cooling Unit

The refrigeration unit is composed of air conditioning equipment strategically distributed in the data center, in such a way that it can provide fresh air to the environment, to the servers and to the communication equipment. This unit is strategically important, because if it fails, it causes an increase in the temperature in the

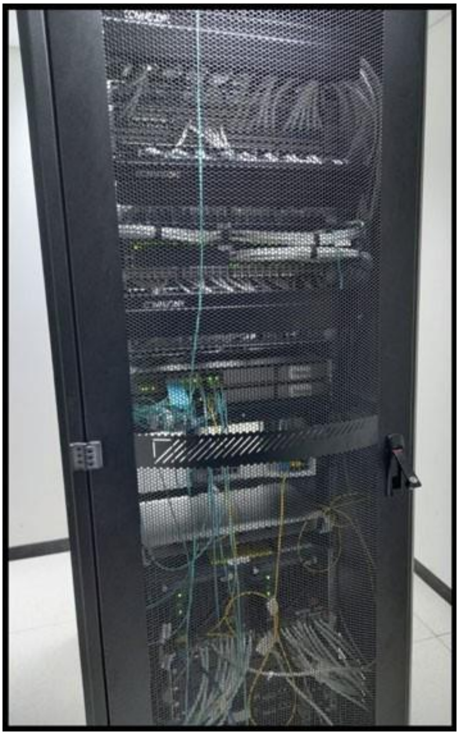


Fig. 12 Front view of the rack with communication equipment

data center, which can lead to failures of the servers, of the communication equipment, and can even lead to a greater failure of the electronic components that they are part of the aforementioned teams.

Among the results presented up to this point, the units corresponding to the information system module have been dealt with, now the units corresponding to the IoT module are presented.

3.6 Acquisition Units

The IoT module acquisition unit is mainly composed of Microchip's PIC-IoT development card, with ATWINC1510 Wi-Fi module, the detail of this module is described in the materials and methods, as a result of this unit we can mention that Through serial communication with the control panel, different data can be recorded, such as the working temperatures of the servers, the temperature in the data center as well as the temperature provided by the air conditioning.

Also through serial communication, the status of the energy being used is sent, we can have information about the commercial line, the substation line and the UPS line, indicating the value of the output voltages, current and working power.

3.7 IoT Units

The IoT unit is also composed of Microchip's PIC-IoT development card, with an ATWINC1510 Wi-Fi module, which fulfills a double function, not only to collect data, but also to provide the information received to a web service that is hosted on the Internet, through the MQTT protocol, which enables the sending of data through a client and a broker, they pack and send the information to the web service, store the information in a database data, which later through SQL queries, reports can be obtained in different formats such as csv, which can then be viewed with Excel programs, Notepad among others. In Fig. 13, you can see the structure and the flow of

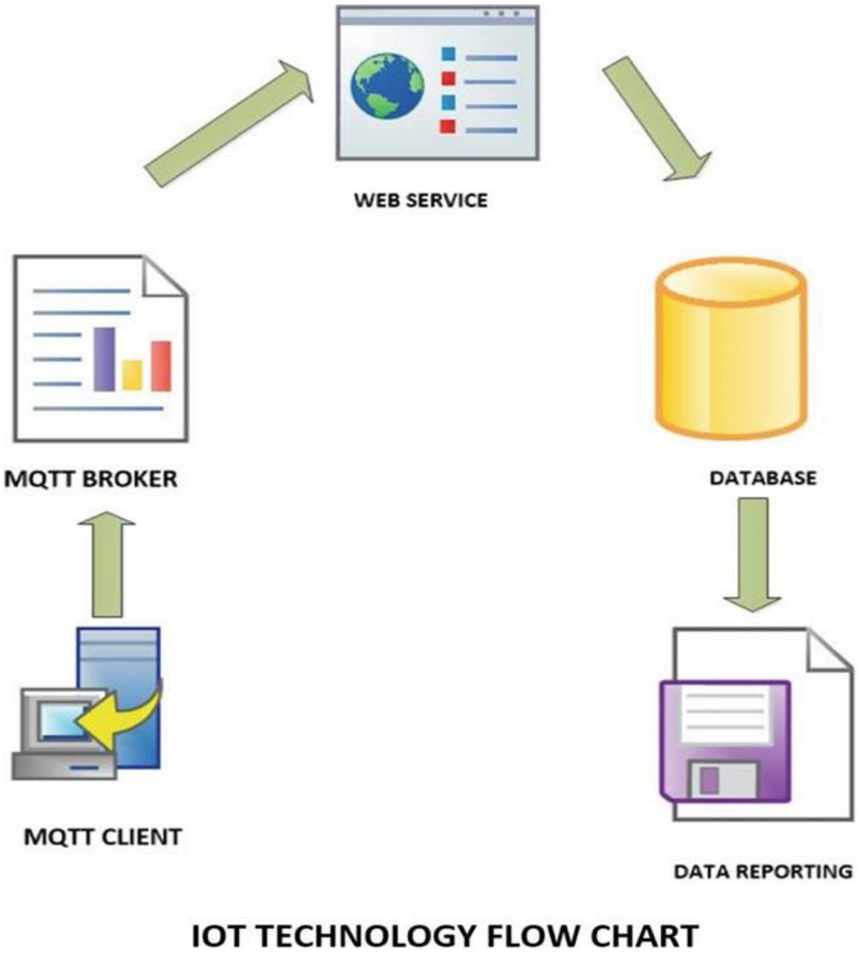


Fig. 13 IoT Unit working mode diagram

information from the acquisition unit to the reports thanks to the IoT-based protocols that are part of the development board.

In Fig. 13, a flow diagram for the storage of information toward the web service is presented, using the MQTT protocol, as well as the storage in the database and the generation of reports.

3.8 Display Unit

The display unit is in charge of being able to present the monitoring data and the control parameters, the application is based on LabView, the display procedure reproduces the information obtained from the web service, this display is carried out constantly and interactive allowing real-time visualization.

The first stage of monitoring The operating parameters that are monitored are:

- Server room, ambient and air conditioning temperatures.
- Voltage, current and power of the commercial line.
- UPS Voltage, Current and Power.
- UPS operation time.

For the control stage, the intelligent current switches are configured and connected to the Wi-Fi network in such a way that they can be activated and deactivated according to the needs and priorities of operation.

In Fig. 14, the display unit is presented with its monitoring and control parameters simultaneously.

The display unit, consists of two components, the monitoring and the control, in Fig. 15, the monitoring module is presented, where the value of the temperatures at each point of the data center is presented, as well as the values of voltage, current and power in the commercial line and in the UPS line, as well as the operation time of the ups, the data present from both the commercial line and the UPS, it is important to carry out the power conversion, it is important to indicate When you want to go from the UPS line to the commercial wing, it must be stabilized, therefore you must know the value of both lines, to change the power line.

The control component is important because it is the one in charge of supplying energy to the servers, this control is carried out thanks to the Smart schich that allows opening and closing the passage of energy to the servers remotely through a web service (Fig. 16).

4 Conclusions

The conclusions that are reached at the end of the presentation of the methodology and evaluate the results are characterized in being able to describe the benefits, functionalities and advantages of IoT technology, and we present the conclusions

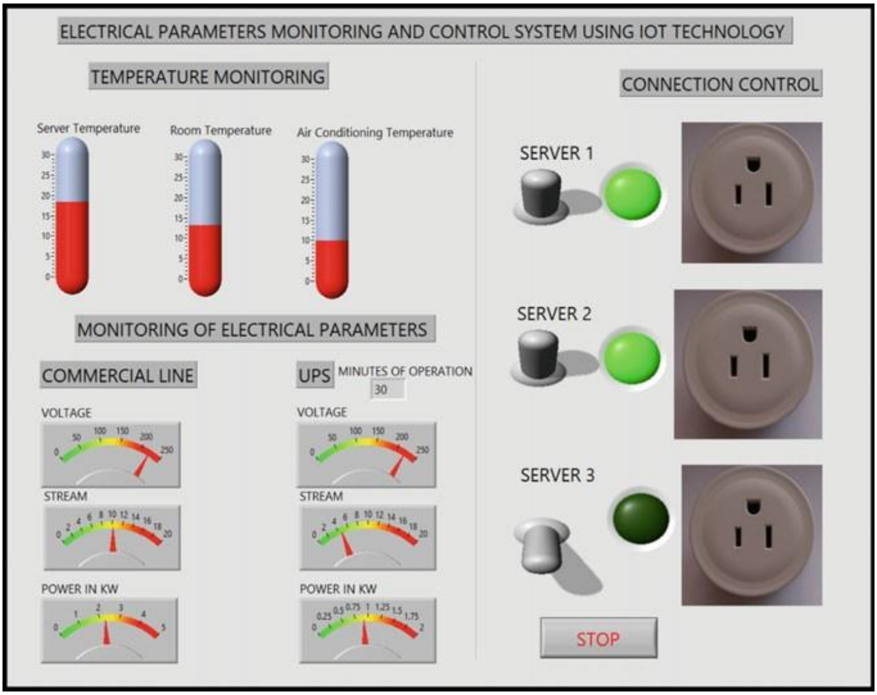


Fig. 14 Application of the display unit in its full format

based on the hardware worked, on the strengths and weaknesses of the hardware used, such as the new opportunities to use them, in energy management to optimize its use, in applications related to information systems.

With regard to hardware on the market, we find many solutions based on IoT technology, the use and application depend on the brand and model, as well as other inputs and functionalities, but they have in common the use of the MQTT protocol, the choice of hardware, it will depend on the scope of the application, in our case the choice is due to the fact that the chosen hardware has an input for serial communication, which is where the information from the control panel is collected through the RS232 protocol, through the port DB9.

Regarding the strengths of IoT technology, we can indicate that it is very helpful in data recording and its subsequent exploitation, in many cases without the need for a computer installed directly to the circuit, the hardware used has a direct connection to the Wi-Fi network, so the information is sent directly to the database, which is connected to the web service, as indicated by the MQTT protocol. One of the disadvantages of IoT technology is that a wireless network is necessarily required to be connected and thus the information can be stored in the Internet cloud, and in most cases, the network requires a Wi-Fi type network.

The new opportunities, allowed by IoT technology, allow the use of multiple sensors in an array-type configuration, capable of being stored in a database in the

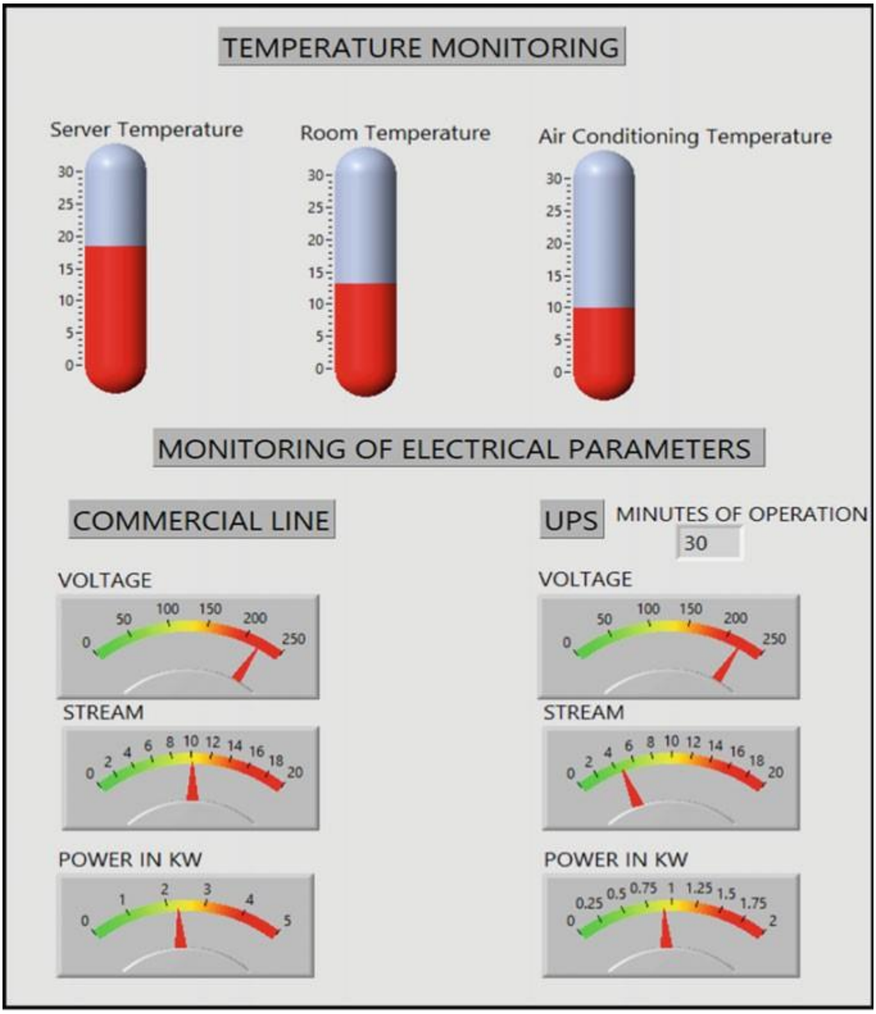


Fig. 15 Display unit application—monitoring component

Internet cloud, these sensors can be placed on different equipment, regardless of the use or application, what must be taken into account is a Wi-Fi connection, depending on the number of sensors, the IoT modules can be used in a cascade configuration, causing the desired solution to be scalable, in order to comply with the design requirements, in addition more information can be incorporated to be stored through the different communication protocols, such as RS232, RS485, parallel communication among others, the use of these protocols will depend on the type of protocol in the IoT circuit.

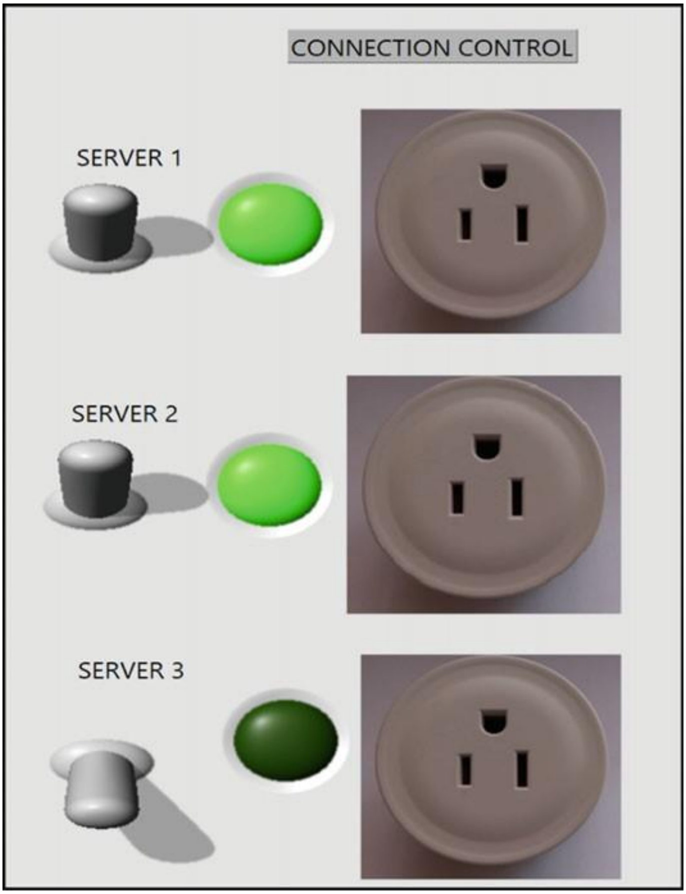


Fig. 16 Display unit application—control component

We can indicate to be able to implement a solution based on IoT technology, it will depend on the scope of the project, for which one of the critical success factors is the choice of the appropriate hardware that meets the requirements and needs in order to comply with the design, the secret lies in choosing the most suitable hardware for the purposes of connectivity and data acquisition channels.

Finally, we can discuss our results with respect to those obtained in works on the control of substations, where the working currents and the current produced by the transformer are monitored thanks to the IoT technology [28], in the work where the monitoring is carried out and the control of the parameters of a transformer where different parameters such as temperature, humidity, ambient light and the operating voltages of the transformer are evaluated [29]. The results obtained in this work demonstrate the application of IoT technology as a mechanism for capturing various physical and electrical parameters, managing to record temperature in the servers, in the server room and that provided by the air conditioning system, as well as the voltage, current and power values of the commercial line and that provided by the UPS, and finally, the operation time of the UPS can be calculated, to be able to make decisions, which will allow services to be turned off by means of intelligent switches.

References

1. Wu G, Wei J, Ye C, Zhong H, Huang T, He H (2012) Specification and monitoring of data-centric temporal properties for service-based systems. J Syst Softw 85(12):2738–2754. https://doi.org/10.1016/j.jss.2012.05.075
2. Chen M, Gao C, Song M, Chen S, Li D, Liu Q (1 Jan 2020) Internet data centers participating in demand response: a comprehensive review. Renew Sustain Energ Rev. Elsevier Ltd. https://doi.org/10.1016/j.rser.2019.109466
3. Alipio M, Tiglao NM, Bokhari F, Khalid S (15 Nov 2019) TCP incast solutions in data center networks: a classification and survey. J Netw Comput Appl Academic Press. https://doi.org/10.1016/j.jnca.2019.102421
4. Dai B, Xu G, Huang B, Qin P, Xu Y (15 Sept 2017). Enabling network innovation in data center networks with software defined networking: a survey. J Netw Comput Appl. Academic Press. https://doi.org/10.1016/j.jnca.2017.07.004
5. Zhang H, Shao S, Xu H, Zou H, Tian C (2014) Free cooling of data centers: a review. Renew Sustain Energ Rev. Elsevier Ltd. https://doi.org/10.1016/j.rser.2014.04.017
6. Faiad AA, Hamdan E, Hamad MS, Abdel-Khalik AS, Hamdy RA (1 June 2020) Differential power processing for data centers applications: a comprehensive review. Alexandria Eng J. Elsevier B.V. https://doi.org/10.1016/j.aej.2020.05.002
7. Liu L, Zhang Q, Zhai Z (John), Yue C, Ma X (1 Nov 2020) State-of-the-art on thermal energy storage technologies in data center. Energ Build. Elsevier Ltd. https://doi.org/10.1016/j.enbuild.2020.110345
8. Tehrani YH, Atarodi SM (2019) Design & implementation of a high precision & high dynamic range power consumption measurement system for smart energy IoT applications. Meas: J Int Meas Confederation 146:458–466. https://doi.org/10.1016/j.measurement.2019.06.037
9. Huang P, Copertaro B, Zhang X, Shen J, Löfgren I, Rönnelid M, Fahlen J, Andersson D, Svanfeldt M (15 Jan 2020) A review of data centers as prosumers in district energy systems: renewable energy integration and waste heat reuse for district heating. Appl Energ. Elsevier Ltd. https://doi.org/10.1016/j.apenergy.2019.114109
10. Shuja J, Gani A, Shamshirband S, Ahmad RW, Bilal K (1 Sept 2016) Sustainable cloud data centers: a survey of enabling techniques and technologies. Renew Sustain Energ Rev. Elsevier Ltd. https://doi.org/10.1016/j.rser.2016.04.034
11. Rong H, Zhang H, Xiao S, Li C, Hu C (1 May 2016) Optimizing energy consumption for data centers. Renew Sustain Energ Rev. Elsevier Ltd. https://doi.org/10.1016/j.rser.2015.12.283
12. Oró E, Depoorter V, Garcia A, Salom J (2015) Energy efficiency and renewable energy integration in data centres. Strategies and modelling review. Renew Sustain Energ Rev. Elsevier Ltd. https://doi.org/10.1016/j.rser.2014.10.035
13. Uddin M, Darabidarabkhani Y, Shah A, Memon J (3 August 2015) Evaluating power efficient algorithms for efficiency and carbon emissions in cloud data centers: a review. Renew Sustain Energ Rev. Elsevier Ltd. https://doi.org/10.1016/j.rser.2015.07.061
14. Jin C, Bai X, Yang C, Mao W, Xu X (1 May 2020) A review of power consumption models of servers in data centers. Appl Energ. Elsevier Ltd. https://doi.org/10.1016/j.apenergy.2020.114806
15. Jin C, Bai X, Yang C, Mao W, Xu X (2020) A review of power consumption models of servers in data centers. Appl Energy 265:114806. https://doi.org/10.1016/j.apenergy.2020.114806

16. Jin C, Bai X, Yang C (10 Dec 2019) Effects of airflow on the thermal environment and energy efficiency in raised-floor data centers: a review. Sci Total Environ. Elsevier B.V. https://doi.org/10.1016/j.scitotenv.2019.133801
17. Zhang H, Shao S, Tian C, Zhang K (1 Jan 2018) A review on thermosyphon and its integrated system with vapor compression for free cooling of data centers. Renew Sustain Energ Rev. Elsevier Ltd. https://doi.org/10.1016/j.rser.2017.08.011
18. Zhan B, Shao S, Zhang H, Tian C (5 Jan 2020) Simulation on vertical microchannel evaporator for rack-backdoor cooling of data center. Appl Therm Eng. Elsevier Ltd. https://doi.org/10.1016/j.applthermaleng.2019.114550
19. Lu H, Zhang Z, Yang L (15 Nov 2018) A review on airflow distribution and management in data center. Energ Build. Elsevier Ltd. https://doi.org/10.1016/j.enbuild.2018.08.050
20. Habibi Khalaj A, Halgamuge SK (1 Nov 2017) A review on efficient thermal management of air- and liquid-cooled data centers: from chip to the cooling system. Appl Energ. Elsevier Ltd. https://doi.org/10.1016/j.apenergy.2017.08.037
21. Ni J, Bai X (1 Jan 2017) A review of air conditioning energy performance in data centers. Renew Sustain Energ Rev. Elsevier Ltd. https://doi.org/10.1016/j.rser.2016.09.050
22. Chu WX, Wang CC (15 April 2019) A review on airflow management in data centers. Appl Energ. Elsevier Ltd. https://doi.org/10.1016/j.apenergy.2019.02.041
23. Zhang K, Zhang Y, Liu J, Niu X (1 Sept 2018) Recent advancements on thermal management and evaluation for data centers. Appl Therm Eng. Elsevier Ltd. https://doi.org/10.1016/j.applthermaleng.2018.07.004
24. Fulpagare Y, Bhargav A (2015) Advances in data center thermal management. Renew Sustain Energ Rev. Elsevier Ltd. https://doi.org/10.1016/j.rser.2014.11.056
25. Gong X, Zhang Z, Gan S, Niu B, Yang L, Xu H, Gao M (15 de junio 2020). Una revisión de las métricas de evaluación del rendimiento térmico en los centros de datos. Edificación y Medio Ambiente. Elsevier Ltd. https://doi.org/10.1016/j.buildenv.2020.106907
26. Matsui K (2017) Proposal and implementation of real-time certification system for smart home using IoT technology. In: Energy procedia, vol 142. Elsevier Ltd, pp 2027–2034. https://doi.org/10.1016/j.egypro.2017.12.406
27. Zhang J (2020) Real-time detection of energy consumption of IoT network nodes based on artificial intelligence. Comput Commun 153:188–195. https://doi.org/10.1016/j.comcom.2020.02.015
28. Zeadally S, Bello O (2019) Harnessing the power of internet of things based connectivity to improve healthcare. Int Things 100074
29. Hossain MS, Rahman M, Sarker MT, Haque ME, Jahid A (2019) A smart IoT based system for monitoring and controlling the sub-station equipment. Int Things 7:100085
30. Hasır M, Cekli S, Uzunoğlu CP (2021) Simultaneous remote monitoring of transformers' ambient parameters by using IoT. Int Things 100390