Reconfigurable data intensive service for low latency cyber-physical systems and IoT communication

Prince Gupta[1], Rajeev Sharma[2], Sachi Gupta[3]
[1]Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ghaziabad, India
[2]Department of Computer Applications, SRM Institute of Science and Technology, Ghaziabad, India
[3]Department of Computer Science and Engineering, Galgotias College of Engineering and Technology, Greater Noida, India

Article Info

Article history:

Received Nov 8, 2023
Revised Dec 9, 2023
Accepted Jan 19, 2024

Keywords:

Cyber-physical system
iFoGSim framework
Internet of things communication
Reconfigurable systems
Xilinx ISE 14.7

ABSTRACT

The fourth industrial revolution is realized through the many developments in cyber-physical systems (CPS) made possible by the widespread use of the internet of things (IoT). CPS sensor networks must enable mobile and wireless CPSs with their specific flexibility and heterogeneity needs without compromising quality of service (QoS). The research article focuses on reconfigurable data communication hardware for numerous IoT-supporting infrastructures and performance estimation using delay, power, throughput, and packet delivery ratio (PDR) for different IoT node configurations. Tree topology-based network configuration from cloud data to sensor fog organizers, sensor network directors, and IoT-embedded sensors is supported. Functional simulation is performed in iFoGSim, Xilinx ISE, and Modelsim 10.0 with a maximum of 64 variable nodes programmed for data communication and interplay verification with a minimum delay of 9.1 ns, maximum frequency of 319 MHz, power of 7.5 mW, throughput of 0.280, and maximum PDR=1. The simulation is applicable for fog computing and CPS processed from different alters in specific topologies.

Corresponding Author:

Prince Gupta
Department of Computer Science and Engineering, SRM Institute of Science and Technology
Delhi NCR Campus, Delhi-Meerut Road, Modinagar, Ghaziabad, India
Email: prince.rkg@gmail.com

1. INTRODUCTION

Industry 4.0 is a new data-obsessed paradigm [1] that is centered on the generation of industrialized information using real-time persistent networks and effective data streams. The fundamental enabling technology for Industry 4.0 is industrial cyber-physical systems (CPS). These CPS make it possible for objects and developments that are in the physical world, such as manufacturing facilities, to be firmly attached and assessed by advanced extrapolative analytics-based machine learning (ML) and simulation models that are in the digital world. The internet of things (IoT) [2] is a major emerging network paradigm that holds the promise of bridging the gap between the physical and digital worlds [3]. It consists of internet-empowered gateways and devices that can sense, gather, receive, and send data. In the context of production, this may entail exchanges with high-definition cameras, radio-frequency identification (RFID) tags, global positioning systems (GPS), and/or controllers and actuators. Naturally, the continuous and pervasive nature of these interactions results in the production of huge data repositories as big data that characterize the operations of the factory. After a sufficient amount of high-quality data has been acquired, these enormous datasets can be analyzed using ML to produce accurate predictions after enough data has been collected.

The most recent advancements in information and communication technology have steered a growth in the number of smart devices that are becoming commonplace in people's everyday lives. These devices aim to improve people's overall quality of life. Medical cyber-physical systems (MCPSs) [4], are becoming increasingly popular in the healthcare industry because they allow for interaction that is both smooth and intelligent computing devices and medical equipment. To help with MCPSs, cloud assets are typically investigated to handle the sensing records from health check devices. However, the high quality of service (QoS) offered by MCPS poses a challenge to the unstable and time-consuming networks that connect the cloud data facility and medical equipment. To tackle this problem, mobile edge cloud computing, also known as fog computing, is developing as a promising potential solution. This type of computing moves the computation sources onto the network edge, such as cellular base stations.

Fog computing, also called edge computing [5], is a reasonably new addition to the field of computing models, intending to provide cloud-oriented services at the edge computing network to support a high number of IoT devices. Fog computing uses a dispersed network of heterogeneous devices, or ‘fog nodes’ to process IoT data near its point of origin, such as Cisco IoT interacting hardware, microprocessor-based datacenters, nanocomputing servers, smartphones, Cloudlets, and desktop computers. Therefore, fog computing is crucial for the reduction in service delivery delay of numerous IoT-enabled systems and dismissing the network of its heavy data load. Fog nodes are not resource-rich like cloud datacenters. Therefore, fog and cloud computing paradigms typically work together to address the resource and QoS needs of large-scale IoT-enabled systems.

An increasing number of powerful end users and devices [6] are being made these days. Some examples are servers and smart access devices, which include smartphones, tablets, smart home appliances, smart traffic route polls on the side of the road, cellular base stations, associated smart vehicles, smart controllers, sensors in smart power grids, smart buildings, and industrial applications-oriented control systems. When operational expertise and information technology work together, many more devices come together at companies that specialize. Most of the use cases come about because of the growth of an industry. For example, temperature-controlled sensors based on the chemical vat and sensors that work at oil rig stations are examples of such data processing and edge computing devices. IoT comes from several different technologies, such as RFID, sensors, and machine-to-machine contact. Massive amounts of data are produced by new applications, and IoT devices are utilized to store, process, collect, and exchange this information [7]. The huge amount of data that is created needs to be handled while working with time-sensitive apps that have strict needs to know where they are, processing data quickly, and using little power. But most IoT devices don't have the power or tools to do all of these things on their own, so they ask servers and data centers that work with cloud computing for help.

Cloud-based and edge-of-network computer nodes work together [8] in the fog to run IoT applications. Most fog computing solutions accomplish this by sending IoT data from its origin to numerous destinations in the cloud and on the network's periphery. If the data can be processed locally at each node along this route, those nodes will accept it. In any other case, the information will be sent to the next node in the chain. The significance of this issue cannot be overstated that the communication latency increases by the time it transfers for the data to be passed rather than accepted to the next node. Instead, it is recommended to adhere to the routing system, which is responsible for maintaining a list of all the nodes that have previously collected data in each environment. The system uses this historical context to send the data to the closest node that regularly accepts data in the same setting. Since fewer nodes along the path need to forward data, the communication delay may be reduced. The utilization of a geographical routing technique [9] has recently attracted a considerable amount of focus in recent times.

The cloud and the fog each rely on resources such as computation, storage, and networks as their primary building blocks [10]. The significance of fog computing, on the other hand, can be determined by several different criteria. Reduced network bandwidth, low latency, mobility, heterogeneity, energy consumption, geographical distribution, privacy, and security are the primary characteristics of fog computing. Different investigators and experts are working in the field of fog computing facing the challenge of coping with the complexity of the system, which is caused by the enormous number of participating devices and the interactions between those objects, as well as the myriad of technologies and applications that are involved. To circumvent these challenges, they often apply models and simulation techniques to get a close approximation of the real fog system. The depiction of the real or proposed system or part of it, in the form of an idealized model, is known as a model. It is used to gain an understanding of the system that is being studied, to understand the many phenomena and the connections among the components, to develop estimations about the performance of the complete system or some of its subsystems, and eventually to entertain them. In situations in which the use of mathematical modeling techniques would be challenging or impossible due to the complexity, scale, or assortment of a fog computing simulation and system frameworks that offer the best solutions in the form of a set of tools that may be used to simulate the system.

2. RELATED WORK

Modeling and simulating CPS behavior [11] requires considering both physical and control factors. Events, states, and responses form the backbone of CPS control. While sophisticated numerical models can now be used to study a system's physics, control is still often handled in a haphazard and limited fashion. Simulations are significant in the domain of CPS. However, it is worth noting that there is now no singular simulator that can adequately mimic the various dependencies of cyber infrastructure. Hybrid modeling methods are seen as a significant rising but immature area of research [12], and the discipline of hybrid modeling itself is expanding within the ML community as well as the scientific community. In recent years, several works have made attempts to combine model-based and ML models to fully take advantage of the potential that the two together offer.

Data provides a direct basis for the overarching paradigm used to uncover CPS [13]. Physical system detection and transition-logic inference are key to this framework's design. It has been successfully implemented in a variety of practical scenarios. The innovative framework's goal is to decipher how CPS work and to use that knowledge to anticipate their future states. Such data is crucial for evaluating the efficiency of CPS, as it can aid in fixing bugs discovered during the deployment phase and directing the redesign necessary to attain the desired efficiency. The most significant shifts are occurring in contexts where CPS [14] are driving disruptive inventions. Because of this development, significant multidisciplinary relationships between enterprises working in information technology and manufacturing are required. These partnerships will improve the links between current ecosystems. Because of digital networks, businesses, consumers, and goods will be enormously interconnected. This will result in a rise in the effects of network possessions and a shared value generation in ecosystems. Manufacturing organizations need to cultivate ties with specialists who are knowledgeable in the areas of sensor technology and networking.

In addition, software businesses can assist in making optimal use of the capabilities offered by CPS. The use of CPS-based technologies like fusion and virtual-real mapping [15], digital twins, technological innovations in virtualization, edge-to-cloud service delivery, and big data in the industrial system create a new paradigm known as the smart factory. Smart factories centered on CPS are defined, characterized, and architecture, with examples of their use in the past provided. The digital twin is a metamodel that combines physical system modeling with cyberspace simulation to generate simulation analysis capability for intelligent solutions. In the pre-production phase of a smart factory, this delivers a wide range of digital designs of production quality, processes, efficiency, cost, and environmental effect studies, all of which can be examined with big data and lead to authoritative answers for the physical space itself.

These technologies, which will be the foundation for future smart services, will strengthen our vital infrastructure and may have a major effect on our daily lives. However, with more CPS adoption comes an increase in vulnerabilities [16] that could have far-reaching effects on users. Designing secure and efficient CPS is a hot topic because of the widespread nature of the security issues in this space. While there is nothing new about the need to secure sensitive information, new vulnerabilities have emerged because of technological advancements. The necessity for novel approaches to CPS security arises from the persistence of both exploitable hazards and cyber-attacks. Any intelligent control system absolutely must have sensors as one of its main components. Information technologies include wireless sensor networks that are developing at one of the fastest rates currently available and hold the potential to have a variety of uses in advanced networks. The Markov model has been offered as a method for conducting reliability studies of sensor nodes in wireless sensor networks. Data has proven that the sensor node reliability depends on the observing strategy and is unimodally related to testing time.

Network on-chip (NoC) is the technology used for CPS, IoT communication, edge computing, and reconfigurable system communication [17]. The NoC communication happens based on 2D and 3D network routers and communication [18]. Wireless sensor network (WSN) research is advancing rapidly in agriculture, industrial, smart grid communication, metering, and monitoring. Zigbee, an IEEE 802.15.4 wireless network protocol, provides minimum cost, optimal power, and low-delay data transfer via personal area network (PAN). Several WSN geographical domain sensor nodes communicate. Through the coordinator node, embedded sensor nodes conduct specialized applications by delivering information into the network. The coordinator collects, stores, processes, and routes data to the target node. Zigbee supports star, peer-to-peer, mesh, and cluster tree topologies. The research study proposes a hardware chip architecture for mesh, star, and cluster Zigbee topologies with 64 nodes communicating [19]. Virtex-5 FPGA pre-synthesizes, and Xilinx ISE 14.7 software simulates 64-bit data transfer as internode communication. The chip's performance is compared using field-programmable gate arrays (FPGA) and timing parameters. The design consideration of recognized hardware design parameters such as memory consumption, and timing parameters such as maximum and lowest period, and frequency support. during the planning stages of a NoC chip design. ML uses decision tree regression, multiple linear regression, and random forest regression to evaluate the accuracy and efficacy of the design. A massively scalable solution to SoC's communication issues, NoC has been proposed. The space, latency, throughput, and power requirements of a NoC all affect its performance. This research investigates the Virtex-5 FPGA's capability of running 2D and 3D mesh NoCs [20]. The Xilinx ISE 14.7 software is utilized for design, and the XY and XYZ routing behaviour models are used for the design of 2D, and 3D mesh NoC, respectively. Modelsim 10.0 is used for the functional simulation. To accurately predict the hardware resource consumption on FPGA, the on-chip communication is tested for both 2D and 3D mesh NoCs of varying cluster sizes. To address the challenges of significant structure in NoC synthesis on FPGA when many processing components [21], routers, and cache controllers are combined with SoC, the technique gives a significant platform to the NoC designers.

WSN and CPS each have their unique routing. Traffic avoidance and congestion control schemes for WSNs include things like optimizing throughput with improved end-to-end data concurrency and parallelism in high-speed WSNs and obtaining node-connected window optimization in distributed fuzzy computing with high-speed transmission control protocol (TCP) in WSNs [22]. Routing protocols are essential in a WSN because they provide efficient and reliable data transfer between the network's two ends, the source, and the destination nodes. The multicast ad hoc on-demand distance vector (AODV) routing has been synthesized for multiple nodes and intercommunication data transfer [23]. The design successfully simulates the hardware device for the suggested routing protocol on a Virtex-5 FPGA using Xilinx ISE 14.7 and the code language very high-speed integrated circuit (VHDL).

CPS use sensors and actuators to connect communication, information, and intelligence to the physical world [24]. A CPS can be made up of several fixed or moving sensor and actuation networks that work with an intelligent decision system. Each WSN has the same issues: network construction, power/ network/mobility management, and security. CPS has cross-domain sensor cooperation, flow of different kinds of information, and smart decision-making and action. Intelligence and knowledge accumulated through many methods of data mining and learning technologies: The data in CPS may contain a high degree of dynamic unpredictability. Technologies like ML and data mining will be essential for retrieving relevant information [25]. Knowing how these sensory data are related throughout time and space is crucial.

3. METHOD AND SYSTEM

Fog computer is a computer paradigm that addresses the issue of high traveling latency, also known as propagation latency, by enhancing the intelligence and resourcefulness of gateways. It is postulated that the gateways possess restricted storage and processing capabilities, which can be employed to handle the data of crucial applications near the terminal devices. This phenomenon contributes to the mitigation of trip latency for the data about important applications [26]. Due to their advanced capabilities, intelligent gateways possess the ability to make informed decisions regarding the processing or forwarding of incoming data. These decisions are based on factors such as the availability of resources inside the gateways and the latency limitations associated with the application data. The fog computing paradigm includes QoS profile, fog infrastructure, IoT application, and appropriate deployment strategies. The model relies on the utilization of the simulator, created in Java, which enables developers to alter the fog infrastructure by defining latency and bandwidth parameters following QoS laws and specifying application requirements. A cost model is available for iFoGSim although it does not provide some features. iFoGSim is a modified version that enables the precise allocation of IoT applications on fog computing platforms. The QoS determination for communication channels is facilitated by the iFoGSim framework, which also encompasses the hardware, software, and QoS configurations of the system. The Monte Carlo method is employed to emulate the characteristics of communication channels, as well as to generate probability distributions for sampling purposes. iFoGSim provides a set of measures that may be used to quantify and compare resource utilization and QoS accuracy.

The components of the fog layer module exhibit a similar conceptual structure to those of the open fog architecture. However, the behavior of edge devices inside this framework differs. Fog computing enables the preparation of data before its transmission to the cloud, hence minimizing communication latency and lowering the necessity for extensive data storage through the application of filtering techniques [27]. In the perspective of IoT, employing this method is generally deemed appropriate for various applications and services. The methodology is focused on the intricacies of fog computing architecture. The edge devices operate autonomously, meaning that each fog node (FN) in the fog layer, consisting of edge servers and edge devices, is responsible for handling a collection of computing tasks. The design includes the following responsibilities: i) collecting data from IoT devices and preparing it for analysis, if necessary and ii) offloading tasks to other edge servers on-premises or in the cloud when the given edge server's resources are insufficient. Due to the critical nature of task completion, task allocation and offloading play crucial roles in data analysis. The methodology of the design is shown in Figure 1.

The fog computing in the CPS system [28] and level processing are shown in Figure 1. The flow of the model is based on the centralized data processing, storage, and analysis that is available in the cloud. The second layer is the fog layer which has the sensor fog organizers [29]. The third layer is the physical layer which consists of the edge node [30] that works with the network directors. In the last level of processing multiple sensors are placed in a sensory environment which can be IoT sensors [31] or some embedded devices.

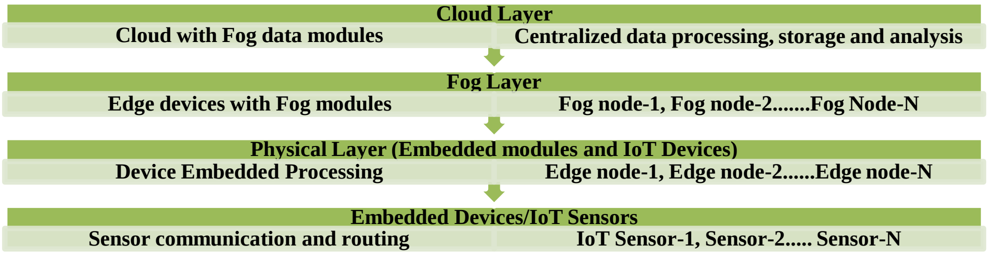

Cloud Layer
Cloud with Fog data modules
Centralized data processing, storage and analysis
Fog Layer
Edge devices with Fog modules
Fog node-1, Fog node-2.......Fog Node-N
Physical Layer (Embedded modules and IoT Devices)
Device Embedded Processing
Edge node-1, Edge node-2......Edge node-N
Embedded Devices/IoT Sensors
Sensor communication and routing
IoT Sensor-1, Sensor-2..... Sensor-N

Figure 1. Fog computing in CPS system and level processing

4. RESULTS AND DISCUSSION

The results are obtained based on the simulation environment of the CPS for multiple IoT sensors in the iFoGSim simulator and evaluated with the necessary performance metrics of the model. The sensors, actuators, and other devices that are based on the IoT make up the physical layer. The information that has been gathered from the various end devices will be sent to the fog layer so that it may be processed and analyzed. To do the data analysis without sacrificing the latency problem is the key purpose of this project. The use of fog computing is the most effective way to meet the requirements of data analytics in an IoT-based environment. Figure 2 shows the iFoGSim simulation environment for 16 sensors. Figure 3 presents the register transfer level (RTL) simulation in Xilinx simulation environment software tool which helps analyze the inputs and outputs pins of the reconfigurable hardware. In the diagram main pins are

Cloud_data <95:0> presents the 96-bit cloud data which is processed through the main sensor environment. Sensor_identification <3:0> presents the 4-bit sensor identification address from sensor-1 to sensor-16 based on their addresses. The Clk is used to provide the clock signal with 50% duty cycle to specify design timing and synchronization in a testbench. A clocking block is a series of clock-synchronized signals. Reset is used asynchronous resets are made by adding them to the sensitivity list to be active on the leading edge of the reset signal.

The performance assessment of the network depends on several factors such as delay, throughput, packet delivery ratio (PDR), and power consumption. The throughput of a system is defined as the number of data units that can be processed within a specific time frame. A wide variety of systems, from organizational structures to computer and network systems, make use of it. Complicating factors in CPS include the need for synchronization and coordination, problems with network dependability and latency, and limited resources for embedded systems. Then delay is an important aspect for the evaluation of the system performance. Numerous power-saving mechanisms take advantage of the reduction in energy consumption of the node peripherals. PDR is used to figure out how reliable a network is by looking at how many of the packets that were sent were delivered correctly.

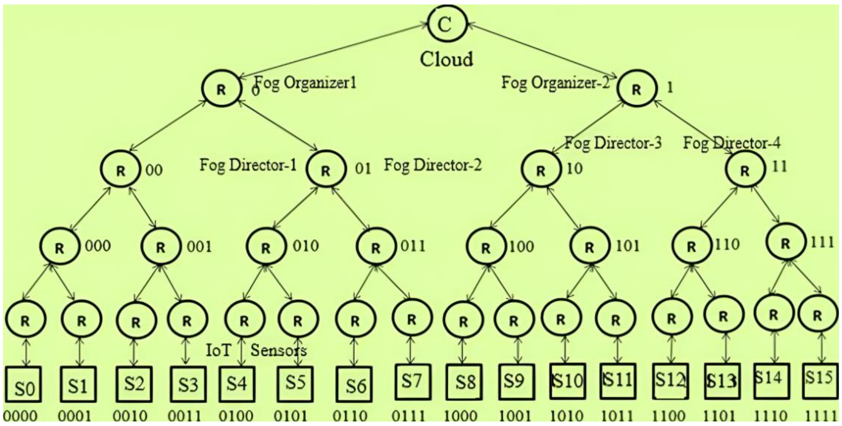


Figure 2. Simulation environment in iFoGSim [22]

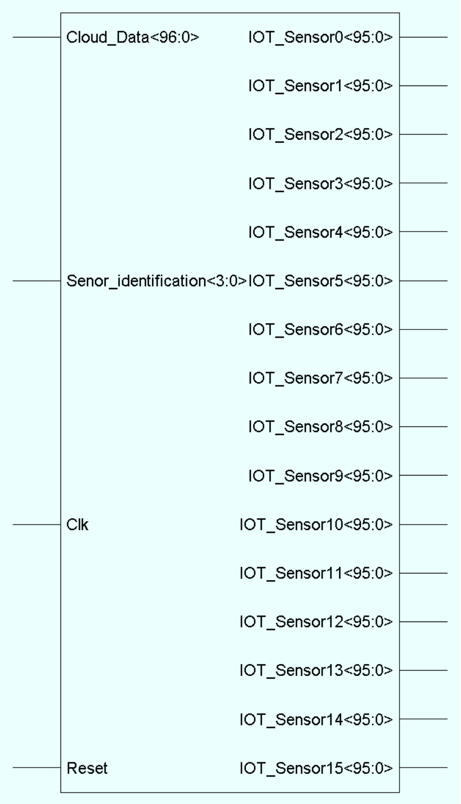
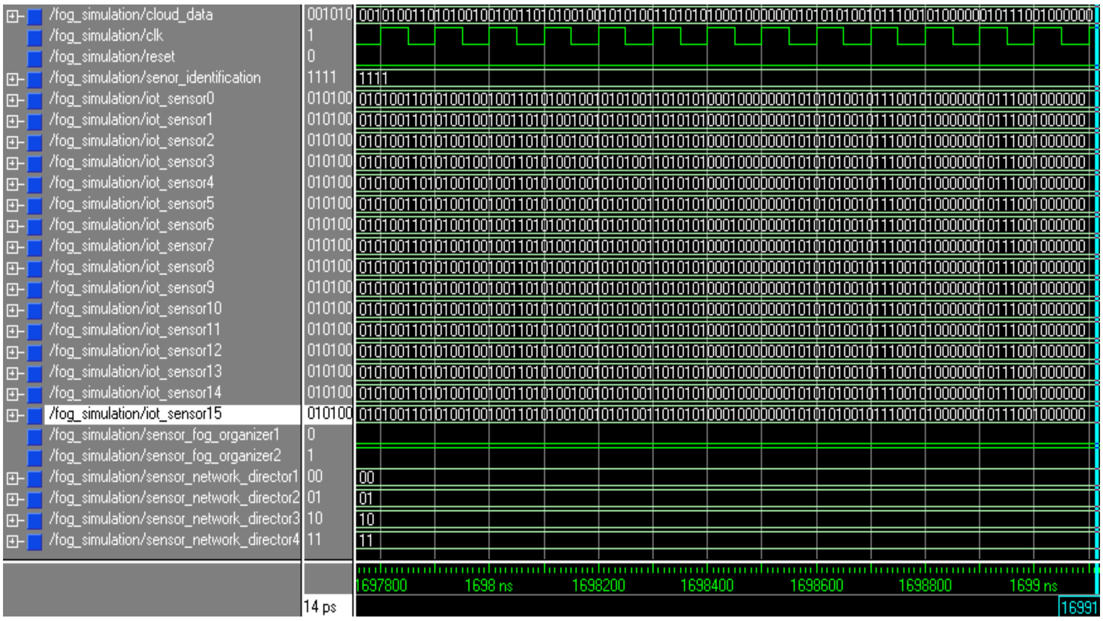


Figure 3. RTL simulation in Xilinx simulation environment

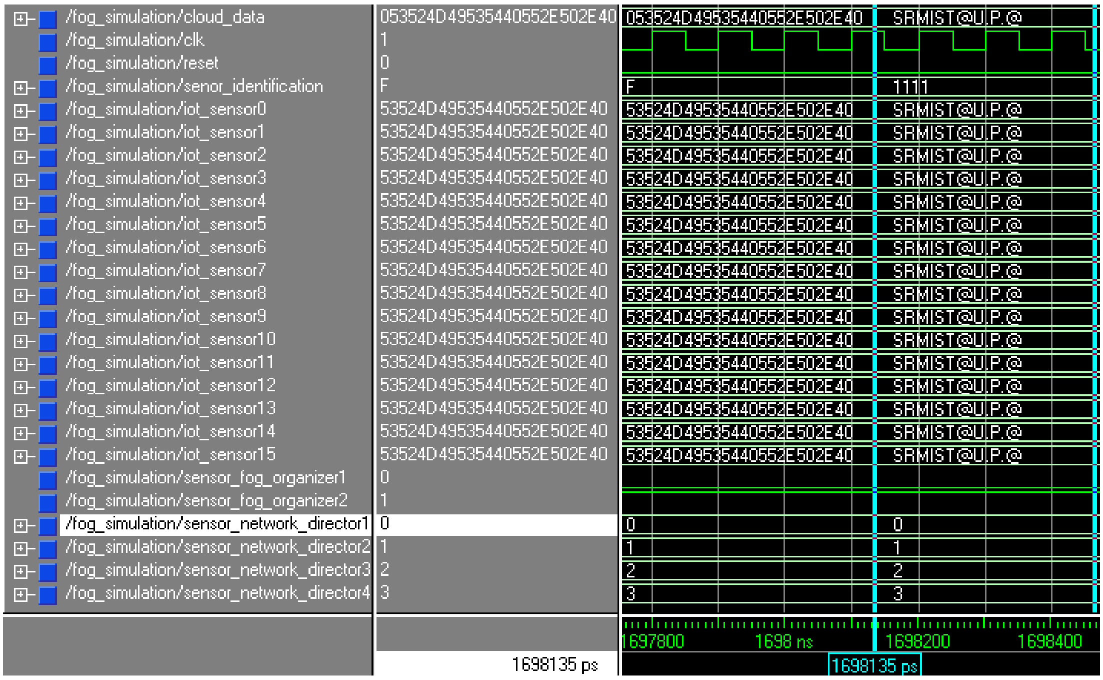
IoT_sensor0<95:0> presents the 96-bit input/output for sensor-0 data communication and processing for sensor_identification<3:0> = “0000”. IoT_sensor1<95:0> presents the 96-bit input/output for sensor-1 data communication and processing for sensor_identification<3:0> = “0001”. IoT_sensor2<95:0> presents the 96-bit input/output for sensor-2 data communication and processing for sensor_identification<3:0> = “0010”. IoT_sensor3<95:0> presents the 96-bit input/output for sensor-3 data communication and processing for sensor_identification<3:0> = “0011”. IoT_sensor4<95:0> presents the 96-bit input/output for sensor-4 data communication and processing for sensor_identification<3:0> = “0100”. IoT_sensor5<95:0> presents the 96-bit input/output for sensor-5 data communication and processing for sensor_identification<3:0> = “0101”. IoT_sensor6<95:0> presents the 96-bit input/output for sensor-6 data communication and processing for sensor_identification<3:0> = “0110”. IoT_sensor7<95:0> presents the 96-bit input/output for sensor-7 data communication and processing for sensor_identification<3:0> = “0111”. IoT_sensor8<95:0> presents the 96-bit input/output for sensor-8 data communication and processing for sensor_identification<3:0> = “1000”. IoT_sensor9<95:0> presents the 96-bit input/output for sensor-9 data communication and processing for sensor_identification<3:0> = “1001”. IoT_sensor10<95:0> presents the 96-bit input/output for sensor-10 data communication and processing for sensor_identification<3:0> = “1010”. IoT_sensor11<95:0> presents the 96-bit input/output for sensor-11 data communication and processing for sensor_identification<3:0> = “1011”. IoT_sensor12<95:0> presents the 96-bit input/output for sensor-12 data communication and processing for sensor_identification<3:0> = “1100”. IoT_sensor13<95:0> presents the 96-bit input/output for sensor-13 data communication and processing for sensor_identification<3:0> = “1101”. IoT_sensor14<95:0> presents the 96-bit input/output for sensor-14 data communication and processing for sensor_identification<3:0> = “1110”. IoT_sensor15<95:0> presents the 96-bit input/output for sensor-15 data communication and processing for sensor_identification<3:0> = “1111”.

Test simulation: Figure 4 presents the simulation for cloud data processing in binary. Figure 5 presents the simulation for cloud data processing in ASCII and hexadecimal. Figure 6 presents the simulation for cloud data processing with different IoT nodes. The test input and outputs are cloud_data <95:0> = 01010011 01010010 01001101 01001001 01010011 01010100 01000000 01010101 00101110 01010000 00101110 01000000 (Binary) = 53524D49535440552E502E40 (hexadecimal) = SRMIST@U.P.@ (in ASCII). Sensor_fog_organizer_1 = ‘0’, Sensor_fog_organizer_1 = ‘1’, sensor_network_director1 = “00”, sensor_network_director1 = “01”, sensor_network_director1 = “10”, and sensor_network_director1 = “11”. Based on sensor_identification <3:0> = “0000” ……… “1111”. The IoT_sensor0<95:0> = IoT_sensor1<95:0> = IoT_sensor2<95:0>= IoT_sensor3<95:0> = IoT_sensor4<95:0> = IoT_sensor5<95:0> = IoT_sensor6<95:0> = IoT_sensor7<95:0> = IoT_sensor8<95:0>= IoT_sensor9<95:0> = IoT_sensor10<95:0> = IoT_sensor11<95:0>= IoT_sensor12<95:0>= IoT_sensor13<95:0> = IoT_sensor14<95:0> = IoT_sensor15<95:0> = 01010011 01010010 01001101 01001001 01010011 01010100 01000000 01010101 00101110 01010000 00101110 01000000 (Binary) = 53524D49535440552E502E40 (hexadecimal) = SRMIST@U.P.@ (in ASCII).



```
/fog_simulation/cloud_data                001010
/fog_simulation/clk                       1
/fog_simulation/reset                     0
/fog_simulation/senor_identification      1111
/fog_simulation/iot_sensor0               010100
/fog_simulation/iot_sensor1               010100
/fog_simulation/iot_sensor2               010100
/fog_simulation/iot_sensor3               010100
/fog_simulation/iot_sensor4               010100
/fog_simulation/iot_sensor5               010100
/fog_simulation/iot_sensor6               010100
/fog_simulation/iot_sensor7               010100
/fog_simulation/iot_sensor8               010100
/fog_simulation/iot_sensor9               010100
/fog_simulation/iot_sensor10              010100
/fog_simulation/iot_sensor11              010100
/fog_simulation/iot_sensor12              010100
/fog_simulation/iot_sensor13              010100
/fog_simulation/iot_sensor14              010100
/fog_simulation/iot_sensor15              010100
/fog_simulation/sensor_fog_organizer1     0
/fog_simulation/sensor_fog_organizer2     1
/fog_simulation/sensor_network_director1  00
/fog_simulation/sensor_network_director2  01
/fog_simulation/sensor_network_director3  10
/fog_simulation/sensor_network_director4  11
                                          14 ps
```

Figure 4. Simulation for cloud data processing in binary



```
/fog_simulation/cloud_data                053524D49535440552E502E40
/fog_simulation/clk                       1
/fog_simulation/reset                     0
/fog_simulation/senor_identification      F
/fog_simulation/iot_sensor0               53524D49535440552E502E40
/fog_simulation/iot_sensor1               53524D49535440552E502E40
/fog_simulation/iot_sensor2               53524D49535440552E502E40
/fog_simulation/iot_sensor3               53524D49535440552E502E40
/fog_simulation/iot_sensor4               53524D49535440552E502E40
/fog_simulation/iot_sensor5               53524D49535440552E502E40
/fog_simulation/iot_sensor6               53524D49535440552E502E40
/fog_simulation/iot_sensor7               53524D49535440552E502E40
/fog_simulation/iot_sensor8               53524D49535440552E502E40
/fog_simulation/iot_sensor9               53524D49535440552E502E40
/fog_simulation/iot_sensor10              53524D49535440552E502E40
/fog_simulation/iot_sensor11              53524D49535440552E502E40
/fog_simulation/iot_sensor12              53524D49535440552E502E40
/fog_simulation/iot_sensor13              53524D49535440552E502E40
/fog_simulation/iot_sensor14              53524D49535440552E502E40
/fog_simulation/iot_sensor15              53524D49535440552E502E40
/fog_simulation/sensor_fog_organizer1     0
/fog_simulation/sensor_fog_organizer2     1
/fog_simulation/sensor_network_director1  0
/fog_simulation/sensor_network_director2  1
/fog_simulation/sensor_network_director3  2
/fog_simulation/sensor_network_director4  3
                                          1698135 ps
```

Figure 5. Simulation for cloud data processing in American standard code for information interchange (ASCII) and hexadecimal

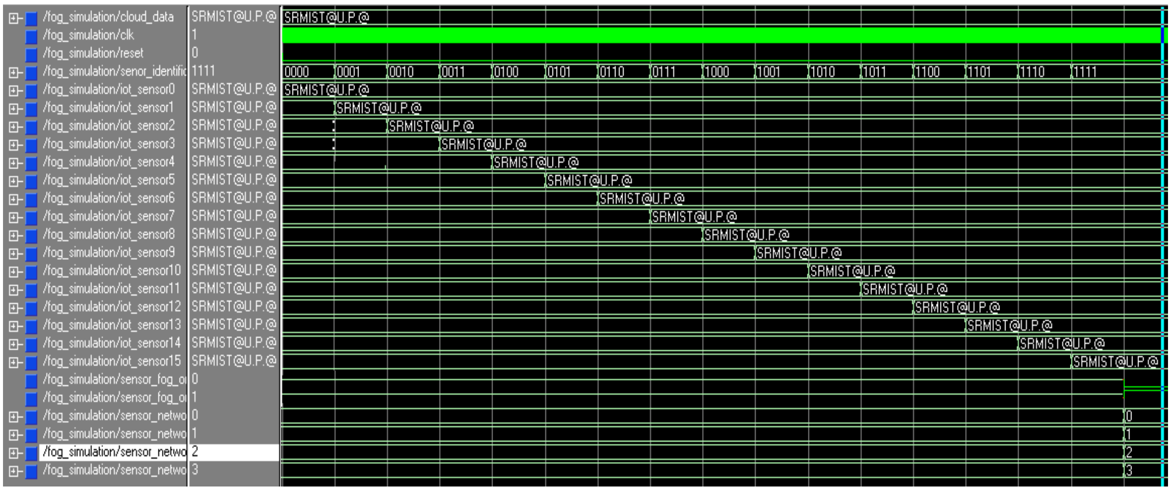

/fog_simulation/cloud_data SRMIST@U.P.@
/fog_simulation/clk 1
/fog_simulation/reset 0
/fog_simulation/senor_identific 1111
0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
/fog_simulation/iot_sensor0 SRMIST@U.P.@
/fog_simulation/iot_sensor1 SRMIST@U.P.@
/fog_simulation/iot_sensor2 SRMIST@U.P.@
/fog_simulation/iot_sensor3 SRMIST@U.P.@
/fog_simulation/iot_sensor4 SRMIST@U.P.@
/fog_simulation/iot_sensor5 SRMIST@U.P.@
/fog_simulation/iot_sensor6 SRMIST@U.P.@
/fog_simulation/iot_sensor7 SRMIST@U.P.@
/fog_simulation/iot_sensor8 SRMIST@U.P.@
/fog_simulation/iot_sensor9 SRMIST@U.P.@
/fog_simulation/iot_sensor10 SRMIST@U.P.@
/fog_simulation/iot_sensor11 SRMIST@U.P.@
/fog_simulation/iot_sensor12 SRMIST@U.P.@
/fog_simulation/iot_sensor13 SRMIST@U.P.@
/fog_simulation/iot_sensor14 SRMIST@U.P.@
/fog_simulation/iot_sensor15 SRMIST@U.P.@
/fog_simulation/sensor_fog_o 0
/fog_simulation/sensor_fog_o 1
/fog_simulation/sensor_netwo 0
/fog_simulation/sensor_netwo 1
/fog_simulation/sensor_netwo 2
/fog_simulation/sensor_netwo 3

Figure 6. Simulation for cloud data processing with different IoT nodes

In the same way, the simulation is carried out for the 64 IoT sensor nodes. The cloud data may vary in size. In that case, node identification will be of 6-bit and sensor_identification <5:0> = "000000" ……… "111111" for sensor-0 to sensor-63. Table 1 lists the assessment parameters of the simulation work. The performance of the system is evaluated based on delay, power, throughput, and PDR. Figure 7 presents the graph for evaluation parameters for CPS.

Table 1. Performance assessment parameters

Nodes/parameters	Dealy (ns)	Power (mW)	Throughput	PDR
2	1.002	0.592	0.192	0.995
4	1.164	1.210	0.210	0.986
8	2.195	1.592	0.168	0.952
16	4.102	2.197	0.269	0.998
32	6.590	4.168	0.237	0.951
64	9.100	7.500	0.280	1.000

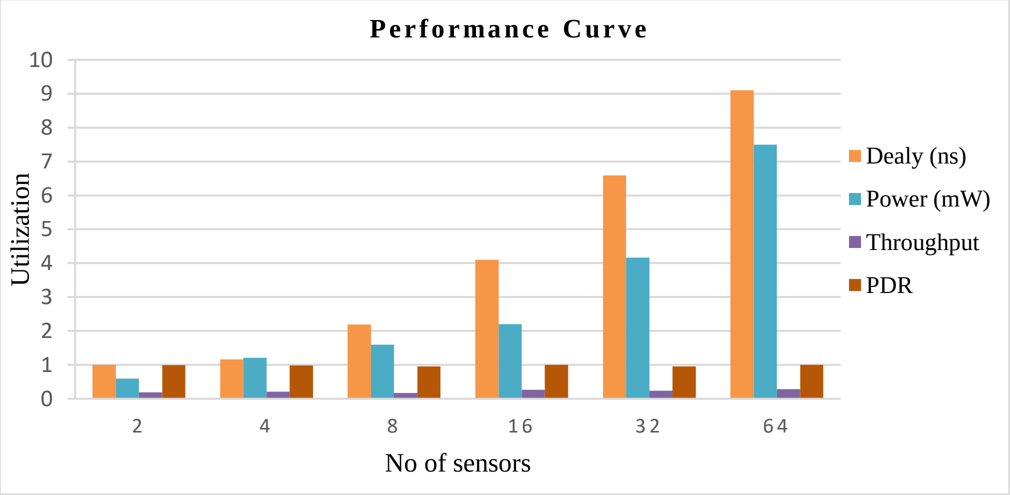
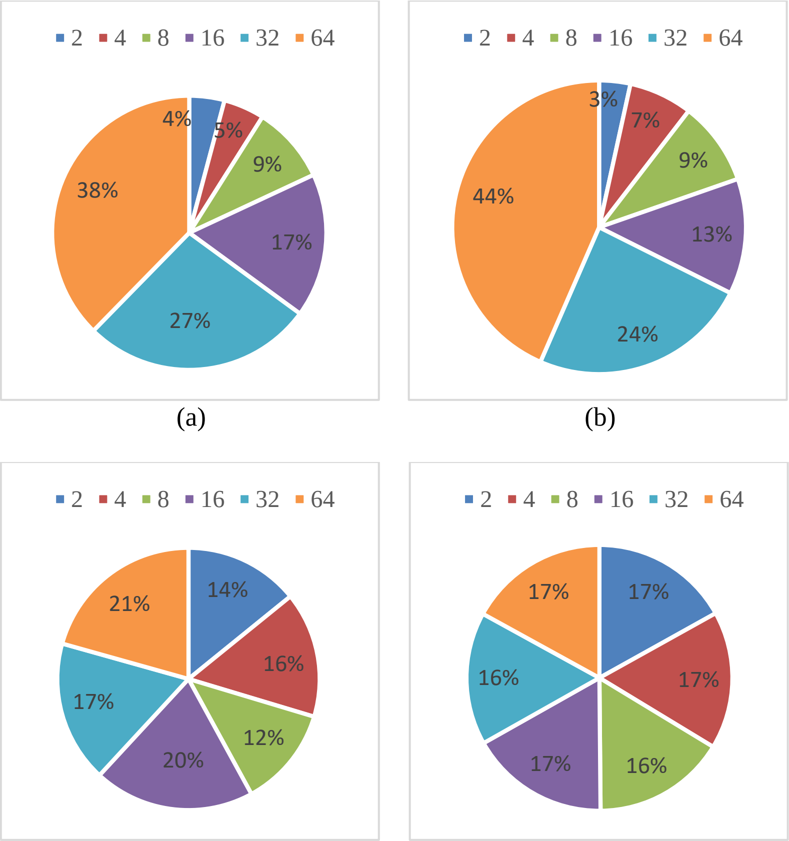


Figure 7. Graph for evaluation parameters

Figures 8 presents the utilization curve for delay on Figure 8(a), power (mW) on Figure 8(b), throughput on Figure 8(c), and PDR consumption on Figure 8(d). The latency (ns) of the CPS system varies depending on the node. The measurements for the latency of IoT nodes are: 1.002 ns, 1.164 ns, 2.195 ns, 4.102 ns, 6.590 ns, and 9.1 ns for 2, 4, 8, 16, 32, and 64 nodes respectively. The power consumption likewise increases proportionally with the size of the nodes, measuring 0.592 mW, 1.210 mW, 1.592 mW, 2.197 mW, 4.168 mW, and 7.500 mW for 2, 4, 8, 16, 32, and 64 nodes accordingly. The throughput varies for each scenario and is measured as: 0.192, 0.210, 0.168, 0.269, 0.237, and 0.280 for 2, 4, 8, 16, 32, and 64 nodes accordingly. The PDR varies depending on the case and is determined by simulation. The values obtained are 0.995, 0.986, 0.952, 0.998, 0.951, and 1.00 for node sizes ranging from 2 to 64 nodes, respectively. In contrast to references [24] and [25], the latency is reduced by 9%, the throughput is increased by 3%, and the PDR =1 for the same node clusters.



(a) (b)

(c) (d)

Figure 8. Utilization for (a) delay (ns), (b) power (mW), (c) throughput, and (d) PDR

5. CONCLUSION

Recent developments in IoT technology and mobile device computing capacity have increased interest in doing computational jobs on IoT or mobile platforms, especially for edge computing solutions. Data transport between IoT nodes and servers must be reduced in edge computing. IoT nodes and mobile devices must compute more. Communication bottlenecks caused by data transfer bandwidth or latency are minimized. The simulation for the reconfigurable IoT communication and CPS was done successfully for 64 sensors. The RTL simulation has shown the configured chip design in Xilinx and functionality is verified in Modelsim for 96-bit data transfer from cloud to sensor nodes. The delay is increasing with the number of nodes increasing from 2 to 64. The delay varies from 1.002 ns to 9.100 ns, power from 0.592 mW to 7.500 mW, throughput from 0.192 to 0.280, and PDR from 0.995 to 1.00. The layer-layer communication is verified.

REFERENCES

[1] T. Fernández-Caramés, P. Fraga-Lamas, M. Suárez-Albela, and M. Díaz-Bouza, "A fog computing based cyber-physical system for the automation of pipe-related tasks in the industry 4.0 phipyard," *Sensors*, vol. 18, no. 6, p. 1961, Jun. 2018, doi: 10.3390/s18061961.

[2] P. O'Donovan, C. Gallagher, K. Bruton, and D. T. J. O'Sullivan, "A fog computing industrial cyber-physical system for embedded low-latency machine learning Industry 4.0 applications," *Manufacturing Letters*, vol. 15, pp. 139–142, Jan. 2018, doi: 10.1016/j.mfglet.2018.01.005.

[3] M. S. de Brito, S. Hoque, R. Steinke, A. Willner, and T. Magedanz, "Application of the fog computing paradigm to smart factories and cyber-physical systems," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 4, Apr. 2018, doi: 10.1002/ett.3184.

[4] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, "Cost efficient resource management in fog computing supported medical cyber-physical system," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 1, pp. 108–119, Jan. 2017, doi: 10.1109/TETC.2015.2508382.

[5] R. Mahmud and R. Buyya, "Modeling and simulation of fog and edge computing environments using iFogSim toolkit," in *Fog and Edge Computing*, Wiley, 2019, pp. 433–465.

[6] M. R. Anawar, S. Wang, M. Azam Zia, A. K. Jadoon, U. Akram, and S. Raza, "Fog computing: an overview of big IoT data analytics," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–22, 2018, doi: 10.1155/2018/7157192.

[7] S. V. Margariti, V. V. Dimakopoulos, and G. Tsoumanis, "Modeling and simulation tools for fog computing-a comprehensive survey from a cost perspective," *Future Internet*, vol. 12, no. 5, p. 89, May 2020, doi: 10.3390/fi12050089.

[8] V. Karagiannis, P. A. Frangoudis, S. Dustdar, and S. Schulte, "Context-aware routing in fog computing systems," *IEEE Transactions on Cloud Computing*, vol. 11, no. 1, pp. 532–549, Jan. 2023, doi: 10.1109/TCC.2021.3102996.

[9] T. Lu, S. Chang, and W. Li, "Fog computing enabling geographic routing for urban area vehicular network," *Peer-to-Peer Networking and Applications*, vol. 11, no. 4, pp. 749–755, Jul. 2018, doi: 10.1007/s12083-017-0560-x.

[10] L. Kaur and R. Kaur, "A survey on energy efficient routing techniques in WSNs focusing IoT applications and enhancing fog computing paradigm," *Global Transitions Proceedings*, vol. 2, no. 2, pp. 520–529, Nov. 2021, doi: 10.1016/j.gltp.2021.08.001.

[11] R. Klein, S. Rilling, A. Usov, and J. Xie, "Using complex event processing for modelling and simulation of cyber-physical systems," *International Journal of Critical Infrastructures*, vol. 9, no. 1/2, p. 148, 2013, doi: 10.1504/IJCIS.2013.051610.

[12] R. Rai and C. K. Sahu, "Driven by data or derived through physics? a review of hybrid physics guided machine learning techniques With cyber-physical system (CPS) focus," *IEEE Access*, vol. 8, pp. 71050–71073, 2020, doi: 10.1109/ACCESS.2020.2987324.

[13] Y. Yuan *et al.*, "Data driven discovery of cyber physical systems," *Nature Communications*, vol. 10, no. 1, p. 4894, Oct. 2019, doi: 10.1038/s41467-019-12490-1.

[14] L. Monostori *et al.*, "Cyber-physical systems in manufacturing," *CIRP Annals*, vol. 65, no. 2, pp. 621–641, 2016, doi: 10.1016/j.cirp.2016.06.005.

[15] G. Chen, P. Wang, B. Feng, Y. Li, and D. Liu, "The framework design of smart factory in discrete manufacturing industry based on cyber-physical system," *International Journal of Computer Integrated Manufacturing*, vol. 33, no. 1, pp. 79–101, Jan. 2020, doi: 10.1080/0951192X.2019.1699254.

[16] Y. Ashibani and Q. H. Mahmoud, "Cyber physical systems security: analysis, challenges and solutions," *Computers and Security*, vol. 68, pp. 81–97, Jul. 2017, doi: 10.1016/j.cose.2017.04.005.

[17] I. Kabashkin and J. Kundler, "Reliability of sensor nodes in wireless sensor networks of cyber physical systems," *Procedia Computer Science*, vol. 104, pp. 380–384, 2017, doi: 10.1016/j.procs.2017.01.149.

[18] P. Agarwal, T. K. Garg, and A. Kumar, "Analysis of 3D NoC router chip on different FPGA for minimum hardware and fast switching," *National Academy Science Letters*, vol. 47, no. 1, pp. 35–39, Feb. 2024, doi: 10.1007/s40009-023-01295-y.

[19] Ompal, V. M. Mishra, and A. Kumar, "Zigbee internode communication and FPGA synthesis using mesh, star and cluster tree topological chip," *Wireless Personal Communications*, vol. 119, no. 2, pp. 1321–1339, Jul. 2021, doi: 10.1007/s11277-021-08282-w.

[20] A. Jain *et al.*, "Smart communication using 2D and 3D mesh network-on-chip," *Intelligent Automation and Soft Computing*, vol. 34, no. 3, pp. 2007–2021, 2022, doi: 10.32604/iasc.2022.024770.

[21] A. Kumar, P. Sharma, M. K. Gupta, and R. Kumar, "Machine learning based resource utilization and pre-estimation for network on chip (NoC) communication," *Wireless Personal Communications*, vol. 102, no. 3, pp. 2211–2231, Oct. 2018, doi: 10.1007/s11277-018-5376-3.

[22] Ompal, V. M. Mishra, and A. Kumar, "FPGA integrated IEEE 802.15.4 ZigBee wireless sensor nodes performance for industrial plant monitoring and automation," *Nuclear Engineering and Technology*, vol. 54, no. 7, pp. 2444–2452, Jul. 2022, doi: 10.1016/j.net.2022.01.011.

[23] M. Yadav, K. Singh, A. S. Pandey, A. Kumar, and R. Kumar, "Smart communication and security by key distribution in multicast environment," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–14, Mar. 2022, doi: 10.1155/2022/1011407.

[24] N. Gupta and A. Kumar, "Study on the wireless sensor networks routing for low-power FPGA hardware in field applications," *Computers and Electronics in Agriculture*, vol. 212, Sep. 2023, doi: 10.1016/j.compag.2023.108145.

[25] N. Gupta, A. Jain, K. S. Vaisla, A. Kumar, and R. Kumar, "Performance analysis of DSDV and OLSR wireless sensor network routing protocols using FPGA hardware and machine learning," *Multimedia Tools and Applications*, vol. 80, no. 14, pp. 22301–22319, Jun. 2021, doi: 10.1007/s11042-021-10820-4.

[26] A. Kumar, P. Kuchhal, and S. Singhal, "Secured network on chip (NoC) architecture and routing with modified TACIT cryptographic technique," *Procedia Computer Science*, vol. 48, pp. 158–165, 2015, doi: 10.1016/j.procs.2015.04.165.

[27] A. Devrari and A. Kumar, "Reconfigurable linear feedback shift register for wireless communication and coding," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 12, no. 2, pp. 195-204, Jul. 2023, doi: 10.11591/ijres.v12.i2.pp195-204.

[28] A. Devrari and A. Kumar, "Turbo encoder and decoder chip design and FPGA device analysis for communication system," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 12, no. 2, pp. 174-185, Jul. 2023, doi: 10.11591/ijres.v12.i2.pp174-185.

[29] K. E. S. Desikan, V. J. Kotagi, and C. S. R. Murthy, "Topology control in fog computing enabled IoT networks for smart cities," *Computer Networks*, vol. 176, Jul. 2020, doi: 10.1016/j.comnet.2020.107270.

[30] A. Kumar, G. Verma, and M. K. Gupta, "FM receiver design using programmable PLL," *Wireless Personal Communications*, vol. 97, no. 1, pp. 773–787, Nov. 2017, doi: 10.1007/s11277-017-4536-1.

[31] F.-J. Wu, Y.-F. Kao, and Y.-C. Tseng, "From wireless sensor networks towards cyber physical systems," *Pervasive and Mobile Computing*, vol. 7, no. 4, pp. 397–413, Aug. 2011, doi: 10.1016/j.pmcj.2011.03.003.

BIOGRAPHIES OF AUTHORS

Prince Gupta is currently working as a research scholar in Department of Computer Science and Engineering at SRM Institute of Science and Technology, Delhi NCR Campus, Delhi-Meerut Road, Modinagar, Ghaziabad U.P. India. He is M.Tech. in computer science and engineering and B.Tech. in computer science and engineering in 2015 and 2006 respectively. He has published three research papers. He is working as an assistant professor, at KIET Group of Institution Delhi-NCR, Ghaziabad-Meerut Road, Ghaziabad-201206, U.P. He is having experience more than 15 years. He can be contacted at emails: prince.rkg@gmail.com or pg8179@srmist.edu.in.

Dr. Rajeev Sharma is currently working as an associate professor in the Department of Computer Applications at SRM Institute of Science and Technology (Deemed to be University u/s 3 of UGC Act,1956), Delhi, NCR, Campus Ghaziabad, India. He received a doctorate in computer science and engineering from Singhania University, Rajasthan. He is the author of more than 30 research publications (Patent, Journals (SCI/SCIE/SCOPUS), and National/International conferences). He is also supervising many students to pursue their research work. He has published a book on wireless applications protocol. He is having experience of more than 17 years. He is also a member of IEEE, Computer Society of India. He can be contacted at email: rajeevks@srmist.edu.in.

Dr. Sachi Gupta is currently working as a professor in the Department of Computer Science and Engineering at Galgotias College of Engineering and Technology, Greater Noida, India. She has more than 19 years of teaching and research experience. She completed her Ph.D. and M.Tech. (gold medalist) degrees from Banasthali Vidyapith, Rajasthan, in Computer Science. She completed her B.Tech. (CS and IT) from UPTU, Lucknow. She has filed six patents, out of which three have been granted, and published more than thirty papers in national/international level conferences/ journals of repute. She is an active member of CSI, Vibha, IACSIT, and IAENG. Her areas of interest include task scheduling, genetic algorithms, machine learning, and fuzzy logic. She can be contacted at email: sachi.gupta@galgotiacollege.edu.