Computers, Materials & Continua
DOI: 10.32604/cmc.2022.023558
Article

Tech Science Press

High Performance Priority Packets Scheduling Mechanism for Big Data in Smart Cities

Fawaz Alassery*

Department of Computer Engineering, College of Computers and Information Technology, Taif University, Taif, 21944, Saudi Arabia
*Corresponding Author: Fawaz Alassery. Email: falasser@tu.edu.sa

Received: 12 September 2021; Accepted: 10 November 2021

Abstract: Today, Internet of Things (IoT) is a technology paradigm which convinces many researchers for the purpose of achieving high performance of packets delivery in IoT applications such as smart cities. Interconnecting various physical devices such as sensors or actuators with the Internet may causes different constraints on the network resources such as packets delivery ratio, energy efficiency, end-to-end delays etc. However, traditional scheduling methodologies in large-scale environments such as big data smart cities cannot meet the requirements for high performance network metrics. In big data smart cities applications which need fast packets transmission ratio such as sending priority packets to hospitals for an emergency case, an efficient scheduling mechanism is mandatory which is the main concern of this paper. In this paper, we overcome the shortcoming issues of the traditional scheduling algorithms that are utilized in big data smart cities emergency applications. Transmission information about the priority packets between the source nodes (i.e., people with emergency cases) and the destination nodes (i.e., hospitals) is performed before sending the packets in order to reserve transmission channels and prepare the sequence of transmission of theses priority packets between the two parties. In our proposed mechanism, Software Defined Networking (SDN) with centralized communication controller will be responsible for determining the scheduling and processing sequences for priority packets in big data smart cities environments. In this paper, we compare between our proposed Priority Packets Deadline First scheduling scheme (PPDF) with existing and traditional scheduling algorithms that can be used in urgent smart cities applications in order to illustrate the outstanding network performance parameters of our scheme such as the average waiting time, packets loss rates, priority packets end-to-end delay, and efficient energy consumption.

Keywords: Packets transmission; scheduling scheme in IoT; software defined networking; big data smart cities; applications for priority packets

1 Introduction

The technology scene has been dominated for years by the term "Smart Cities", which seek to provide a digital environment that stimulates learning and creativity that promotes a sense of happiness and health. There is more than one definition of this term, and sometimes more than one designation, such as "Digital Cities" and "Ecological Cities", and they disagree according to the goals set by those responsible for developing them [1].

The term smart cities and big data have long been of particular interest in the world of technology, and the use of these technologies is no longer a matter of technical luxury in many fields, but rather a necessity that is required by the need to control devices and equipment remotely based on the data that been recorded immediately on the one hand, and the need to process huge data that helps to make decisions automatically and instantly on the other hand. The merging of these two terms has led to the creation of new applications in promising smart environments, which are based on solving problems, facilitating matters, and enhancing the decision-making process, so that smart cities collect information between information technology and operational infrastructure, and big data plays its role in managing and analyzing large data with high-speed via sensors, applications, and devices to collect, share and transfer data in real time [1].

In order to manage this data, there is a need to distinguish three levels: collection level, management and processing level, and application level, which this paper [2] focuses on data processing level that aims to analyze and manage the data generated by the collection level before archiving by representing the design and implementation of a smart city Big Data Processing platform.

The IoT and big data technologies play an important role in the development of smart cities. The Internet of Things allows to connect disparate devices and communicate over networks and data that is potentially meant to be shared. IoT and Big Data technologies help smart cities to manage large amounts of data generated using technologies like Hadoop, Spark, Hbase and big data analytics tools like machine learning, data mining, data analytics, etc. The information collected has a great impact in various areas of the smart city such as security to ensure the safety of people; Transportation to reduce traffic jams; Urban planning identifies areas in need of improvement and modernization; Sustainability to provide sustainable development needs to be renewed [3].

The integration of the Internet of Things and Big Data has become the backbone of smart city initiatives to connect human resources, social capital, and ICT infrastructure to respond to public challenges, achieve sustainable development, and raise the standard of living of individuals [4].

In general, "Smart Cities" work on developing the economic and social levels and their purpose is to provide an environmentally friendly digital environment that stimulates learning and creativity so that if you are able to carry out your business and carry out your activities from your current place based on technology, which in turn helps improve the efficiency of resource consumption and improve the standard of living, the availability of good and fast services, easy mobility, and a safe and less polluted environment. All these are indicators that you live in a city that has the main characteristics of smart cities. New technological developments have helped governments realize the dream of smart cities, as developments in the Internet of things based on connecting devices, exchanging data and launching central control systems have contributed to saving energy consumption and improving the traffic management system in addition to artificial intelligence applications that analyze a large database of data to enrich the process of making the decision, and has already been relied upon in many areas, including the environment, agriculture, financial institutions, health and educational facilities [4].

‘Smart city’ has various definitions. Some are tech savvy; others emphasize the ideological aspect. However, at the heart of the concept is the desire to use technology to better serve cities and urban communities [5]. With the rise of the smart city concept, information technology and interdisciplinary approaches aim to help improve the use of city services and infrastructure, balancing the sustainability of social, economic, and environmental impacts. Technological cities apply information technology and advanced analytical tools to provide citizen-oriented services rather than having citizens be at the center of change that the main beneficiaries of the new city paradigm need to make. You need to develop an approach. To begin the transition to smart cities, we need to establish a metric model to set goals to improve the quality of life of our citizens [6].

Smart cities is a new concept for inserting the technology in environment using different electronic devices, sensors and interaction methods between terminals for the purpose of collecting data. These data are used to manage resources and provide services to end users [7]. Smart cities contains multiple components starting from infrastructure elements (e.g., smart transport networks, smart office facilities, smart houses and etc.) or behavioral elements (e.g., smart politics, smart people, smart living and etc.) [8]. It is an integration between communications, information technology and smart devices in order to build IoT networks that connect people with each other or facilitate providing end users services. Applications of smart cities are developed in order to deal with real-time responses. For example, sending data packets from a source node to a destination node to inform hospitals or police stations about emergency in specific area [8]. In other words, IoT is defined as the idea of connecting various physical devices to the Internet, as well as the concept of computers that express the ability of each device to identify itself with respect to other devices to form a virtual network that combines several distinct electronic devices, software, and sensors to connect them via the internet. These things can exchange data with each other [9].

Originally, the main idea of the IoT is to interconnect multiple analog and digital electronic devices so that information can be communicated efficiently. IoT is a growing field, and it is something that is growing exponentially over the next few years [9]. Despite the simplicity of this concept, the world of IoT is very complex and difficult to define in a specific sentence. On a technical level, it means the process of collecting huge amounts of random data from various things connected to the Internet, analyzing it, and then predicting its behavior in the future. This includes many protocols and programming languages. Thanks to IoT, most electronic devices will become autonomous and do not need any human intervention. These devices will be able to report their malfunctions and fix them on their own, starting with self-driving cars, lighting system in the home, and other various IoT application.

So, in one of IoT applications, if a person is watching TV, and a sensor installed in the refrigerator finds that the water bottle in the refrigerator needs to be refilled because the water level has get down, the refrigerator will turn on all the sensors that installed in the rooms of the house to detects that if there is someone in the house who can go to the refrigerator and refill the water bottle, conveying the information by broadcasting the message “Search” to all of these sensors that have close to a transmission range of 8–10 meter, which overlaps each other, allowing them to freely communicate and share messages with each other. Also, The IoT can be very useful in solving the problems of people with physical or special disabilities in their daily lives [9].

The IoT field is a comprehensive concept that includes integrating several devices, software and networks together to obtain the desired results and achieve the optimal meaning of IoT. Which an integrated IoT system includes four main components which are sensors (or any other type of physical device), Internet connection, data processing software and a user interface.

One of the features of IoT is that it has the potential to accelerate the 'sharing economy'. So, by introducing new technologies for managing and tracking small objects, new auxiliary and economical items can be shared outside of society, airplanes, cars and motorcycles. As the trend continues, it exclusively introduces new applications that drive new business models and revenue prospects. It takes devices and sensors to a more precise level, enabling the creation of new uses, new applications, new services, and new business models that were previously uneconomical. It is also dangerous for many established industries. According to Gartner's chart today, it is one of the top 5 IoT technologies in the world. This means that it is widely used in a variety of roles in various fields such as smart home, vehicle tracking, monitoring of children and the elderly, and daily work. However, the reality is that the sector currently employs many IoT-enabled devices [10].

As for the mechanism of IoT, it starts with sensors that start collecting data from their environment in which they are located, and then this data is sent to the Cloud, which is a huge network of super servers that provide different services to individuals and companies, where the sensors are connected to the servers in different ways that may constitute: Smartphones, satellites, Wi-Fi, Bluetooth, etc. Once the data reaches the cloud, it is processed using a data analysis software. The process of processing this data may be simple or complex, depending on the amount and type of data obtained. Finally, the results are sent to the end user in the form of a specific alert, so that the user changes or adjusts the sensor settings, and sometimes the sensors are modified automatically without the need for human intervention [10].

Big data is a collection of large and complex data sets that are difficult to process using data management tools. Assignments include capturing, curating, storing, retrieving, sharing, sending, analyzing, and visualizing. Big data is moved more frequently and at larger sizes across networks across the four dimensions of volume, speed, diversity, and reliability [11]. It is huge data that cannot be processed by conventional methods because big data analysis and processing takes time, effort and huge cost technicians have been forced to rely on artificial intelligence systems with the ability to learn, reason and process. In addition to using cloud computing technology to complete tasks, it uses complex algorithms to respond to situations that are not programmed into the computer.

Big data provides a deeper understanding of customer requirements, so if it can be processed profitably, it provides the enterprise with a highly competitive advantage and makes the right and appropriate decisions within the enterprise. Based on the information extracted from the customer database, it will improve efficiency and profit in a more effective way and reduce losses. Therefore, scientists regularly face limitations due to large datasets in many areas, including genetics, phylogeny, meteorology, complex physics simulations, biology, and environmental studies. All of this is huge and fast data. Some workflows require the ability to transfer a single 100TB dataset within hours. In today's packet networks, data is transmitted over the network as a set of packets that are transmitted one at a time without looking at the entire data with associated Quality of Service (QoS) requirements. Therefore, it is difficult to provide QoS for bidirectional big data applications, the usefulness of the network is low, and the enhancement of the forwarding protocol is very urgent. Whether optical networks operating in circuit-switched patterns have a new role for this [11].

Big data is receiving widespread attention in developing business applications such as IoT, and one of these service-based services is the efficient delivery of entity data that is currently designed based on end-to-end communication on the Internet [12]. That said, most of them are implemented on a central server/cloud, and big data from remote servers/clouds should probably be distributed to users via the same route [12]. Therefore, the current big data provisioning is highlighted by massive redundant traffic and significant latency. Network design of big data is essential to support the provision of large amounts of big data, and an architecture that supports data provision in networks such as Information-Centric Network (ICNs) [13] and Named- Data Network (NDNs) [14] is required. It is suggested for content-oriented applications. These networks reduce latency and excessive bandwidth consumption near users where data is cached on intermediate physical entities (IPEs) such as routers.

The processing of big data currently requires massively parallel processing on thousands of servers, all of which must be directly connected to each other. With the advent of large data sets, there is an ongoing demand for additional data exchange capacity in data centers. Ultra-wideband data center network operators face the daunting challenge of scaling their networks (bandwidth) to a hitherto unimaginable size, maintaining a seamless connection between any two points in their network.

In current data networks, data is a packet from an application before it enters the network and passes on the network packet by packet until all the pieces reach their destination. This simplifies the interaction between applications and the network, as packet processing and transport are the only functions that the network must perform, but it also comes with the disadvantage of increasing the amount of traffic and data [15]. Application architects find it difficult to move big data from one side of the network to another, especially in the context of big data. Without the use of concurrent TCP streams or other transport-level improvements, moving gigabytes of data files across congested networks is difficult, if not impossible. Additionally, it is difficult for network operators to deploy new services over shared packet-switched networks, which always experience unpredictable congestion [15].

The rest of this paper is organized as follows. In Section 2, we discuss some related works regarding the scheduling algorithms in big data smart cities. In Section 3, we give our problem statement. Section 4 details our proposed Priority Packets Deadline First (PPDF) scheme with Software Defined Networks (SDN) technique, and explains all related algorithms (i.e., connecting the source nodes with the SDN controllers algorithm, filtering data packets algorithm, and forwarding priority packets algorithm). Section 5 illustrates the simulation results and evaluates the performance of the proposed PPDF scheme with SDN technique and compare PPDF with existing well known scheduling algorithms utilized in big data smart cities. Finally, we conclude the paper in Section 6.

2 Related Works

The term IoT refers to the network of devices capable of collecting and sharing data with other devices on the same network. This allows objects to be sensed and controlled remotely through the existing network infrastructure. When the devices are able to communicate with each other, this leads to Creating a platform where automation can be programmed, for example a smart house is one of the best examples of how the Internet of Things can be used well, where every piece of office equipment can be monitored, whether it is smart locks installed on every door, a smart coffee machine or a unit Air conditioning or smart refrigerator, controlled remotely. All of these services need to be scheduled in several ways that will be based on reducing costs, improving safety and automation, improving workers' efficiency, and also reducing the level of electricity consumption and improving the mechanism of communication of devices with each other.

IoT is about designing optimal scheduling algorithms, which should maximize CPU usage and throughput and minimize latency and power consumption [16]. In general, IoT applications can shorten the life of the network due to the high sensing cost of repeatedly performing tasks on various sensor nodes. Some approaches can solve this problem by assigning similar tasks in a specific area to a single machine. However, choosing to run a single machine was a difficult matter. In addition, efficient scheduling of tasks prevents repetition of the same task and leads to unnecessary inter-node communication [17].

2.1 Priority Based Adaptive Scheduling Algorithm for IoT Sensor Systems

Authors in [18] proposed a priority based adaptive scheduling algorithm (PASA) for IoT sensor systems which is the traditional scheduling approaches do not meet the various quality of service (QoS) requirements and requirements of heterogeneous applications. The proposed PASA takes into account the requirements of heterogeneous applications, such as data rate, minimum delay transmit power, remaining energy, and remaining buffer size of the device [18]. The proposed algorithm considers IoT sensors with different applications. The base station allocates a collision-free time slot to each node based on the priority of the traffic. The duty cycle (ST) of each node is adaptively fixed based on the remaining buffer size (RBS), the remaining energy (RE), and the required transmit power (TP) of the priority queue level of the traffic.

2.2 Scheduling Services on an IoT Device Under Time-Weighted Pricing

The authors presented an IoT network device that contains an interface with one specific computational resource available. Efficient use of available IoT resources improves the quality of service (QoS) of IoT networks that serve smart cities [19]. All resource allocations must be completed within a given fixed window, and all services are parameterized by a price weighting function to indicate the margin of error provided at the start of a given window.

At the same time, they provided a method of mathematical optimization to minimize the total cost of allocating all demand in the scheduling window, taking into account the tolerance level of each service. It also proves that their problem is computationally difficult and provides numerical results to gain insight into the impact of the various price weighting features of the allocation distribution within the scheduling window.

Several sets of simulations were run to evaluate how using a different set of price weighting functions affects the distribution of allocations over time. If the mixture consists of tolerant services over generous services, the former takes precedence and thus at least most of the former are provided as early as possible. Conversely, if the mixture consists of tolerant services rather than intolerant services, and if the needs of the latter are fully met at the beginning of the reservation window, then, the latter will be fully served as soon as possible. Overall, the numerical results demonstrate the ability of the formulation to model acceptance levels of smart city services.

2.3 Energy Efficient Scheduling for Networked IoT Device Software Update

This system proposed a mechanism to schedule updates via the IoT network to minimize energy consumption while meeting the constraints of the deadline for updating all devices. It mathematically formulates the energy efficiency update scheduling problem as an optimization problem using a new energy model of the update process and propose an algorithm to estimate the optimal schedule for updating all devices in the network. And by examining the proposed algorithm on three network instances, including the tree-part-mesh-full-mesh topology. Simulation results show that this algorithm can achieve near-optimal values. This is a difference of only 3.2% from the minimum in the best case [20].

It focuses on the IoT network model, which includes many connected IoT devices and gateways. All gateway devices are considered “nodes” in the chart. The device receives the component at both the gateway and other devices. You can download from multiple nodes at the same time or send to multiple nodes, but you can only download up to one component on one node at a time. They proposed an algorithm called ESUS, which uses step **P1** to create an energy efficient schedule without taking into account the time limit T_{max}.

2.4 Optimal Resource and Task Scheduling for IoT

Based on [21] task scheduling poses a number of challenges in IoT. Scheduling work in an IoT-aware environment uses sensors and RFID in particular, so resource usage needs to be reduced. And should be a method that used to extends the life of the network and reduces communication between nodes. And the authors find that by combining the non-priority, non-occupancy algorithm and Ant Colony Optimization (ACO), they proposed a hybrid resource and job scheduling algorithm that finds the optimal path to guide the sensor node in order to prioritize and execute the work. Here, there are no real-time requirements, and you can focus on the tasks that you can save to the job queue of your IoT device and perform later. The proposed algorithm consists of two types of modules:

2.4.1 TPP (Task PreProcessor)

The preprocessing of individual tasks based on user queries is executed by the first module Task PreProcessor that works within the IoT gateway. The main role of an IoT gateway is to create a workgroup and identify the virtual object that leads the workgroup. Creating a task group requires a task preprocessing that the task preprocessor does. The TPP is responsible for splitting user queries into multiple tasks and specifying the appropriate sensor to perform the task [21].

2.4.2 TPS (Task Path Selection)

The virtual leader node is responsible to lead the workgroup. Therefore, it keeps track of all the nodes that have joined the workgroup. Therefore, it is the virtual reader's responsibility to choose the route that forwards the task's packets between task groups to complete the task. The virtual reader makes a work path selection to reach the sensor node determined using the ACO algorithm. Issues that cannot be negotiated are directly assigned to the TPS, issues that can be negotiated are divided into groups, and only the issues of the leader of each group are assigned to the TPS. TPS schedules tasks based on delay parameters. Jobs with the lowest time delay are reserved first from the run queue, followed by scheduled tasks with the highest delay [21].

2.5 Scheduling Method for Collaborative Detection Traffic in the Bottom of IOT

For the purpose of fast scheduling of joint detection traffic in IoT, a joint detection scheduling model on the bottom of the IoT is designed in this model [22]. The model prioritizes the detection layer traffic data through a local scheduler in the collaborative layer, then schedules the traffic data using an enhanced weighted round robin (IWRR) method, and the IWRR method optimizes real-time and non-real-time mixed traffic scheduling borrowed priority. Data by building a temporary queue in the rank. The simulation results show that, compared with the conventional WRR method, the IWRR method can transmit real-time traffic data continuously and timely, while the average transmission delay of the traffic data is low.

Many cities are now seeking to make use of Big Data, and emerging technologies such as: Artificial Intelligence, Machine Learning, IoT etc., to facilitate the daily lives of residents. So that there is a relationship between LoT in smart cities and big data. LoT can tackle most of the challenges related to big data. Since the technologies of LoT will spread within most sectors, this will lead to the flow of very large volumes of data and new methods will be established to collect this data, analyze it and benefit from its information.

2.6 Big Data Processing Platform for Smart City

According to the paper [23], smart cities have vast amounts of value information and visions extracted from big data generated by sensor applications, devices and people. Creating big data modeling is essential to overcome this problem. Paying special attention to creating a multi-layered smart city approach, the authors will propose a big data processing design for a smart city that uses the collected data by adding a data modeling approach that handles both transmissions. The authors are interested in data management and processing layer using big data processing tools and big data transmission tools. Then, after the process of comparing the effectiveness of the tool and the results from various aspects, the platform requirements are determined and the proposed structure is connected to the proposed structure of a data processing platform Smart City with many advantages such as reliability, scalability, high error etc., resistance and high efficiency.

2.7 Network Experience Scheduling and Routing Approach for Big Data Transmission in the Internet of Things

According to the authors [24] recent advances in Internet-related technologies in things have led to the transition to smart applications such as smart cities, smart homes, and online applications for business management. This, in turn, added significant additional load to the existing network infrastructure. Also, these applications use big data and require relatively short response times. Therefore, the researchers have introduced new scheduling and routing mechanisms to improve the end-user experience and provide increased throughput for big data applications. The results obtained demonstrate the efficiency of this new mechanism over other competitive methods in terms of delay in the network, network throughput and packet deletion.

They proposed new centralized routing and scheduling algorithms and evaluated the new approach using extensive simulations with other centralized routing and scheduling algorithms, resulting in better quality of service (QoS) measurements than traditional methods. It shows excellent performance. The results also show that the performance of the entire network, which uses the previous network environment to determine the routing path of packets, can be significantly improved. Large-scale applications, such as IoT heterogeneous networks, high-load networks, network environment approaches can use network resources to improve network performance and have a significant impact on end-user experience.

2.8 Big Data in Motion: A Vehicle-Assisted Urban Computing Framework for Smart Cities

According to the vast number of these devices used in smart cities, the efficient management of IoT resources and the data generated by these resources is one that has brought about unprecedented growth in big data. Causes previously related capacity issues on the device. Existing networks are not sufficient to support this huge amount of data transfer, so the authors [25] proposed an efficient data transfer frame based on volunteer vehicles which used to carry data in the direction of their destination. This framework promotes independence, social awareness and energy saving through urban computing. Also allows the research community to easily benchmark their own path selection algorithms. And the result is that the energy consumed by the vehicle is significantly less than the energy consumed by transmission over the Internet; thus, reducing carbon dioxide emissions, which demonstrates the efficiency of the proposed data transfer framework. The results also provide some insight into the optimal configuration of the vehicle's data transmission network, based on latency, energy consumption, and data spot utilization analysis.

Software-Defined Networking (SDN) is a new concept that transforms the way networking hardware works with software and transforms networks toward virtualized services. Technological advances in virtualization services, distributed systems architecture, big data, and cloud computing require fast and resilient networks that adapt quickly when using centralized smart technologies and SDNs are changing both the role of networks and where the control is located. SDN technologies can be deployed on virtual machines that can be installed on servers, such as VMware NSX, or run-in new switches that accept advanced network operating systems such as Big Switch or Cumulus.

Because IoT devices are internet-based and contain sensitive information, security concerns are being raised and it needs to find ways to improve security between these types of devices (SDN) is a promising computer network technology that introduces a central program named 'SDN Controller' that enables network-wide control. Therefore, using SDN is a solid solution for improving IoT networking performance and overcoming the shortcomings that currently exist [26].

2.9 CORAL-SDN: A Software-Defined Networking Solution for the Internet of Things

According to [27], SDN solutions have been provided for wireless sensor networks (WSNs) that will use an intelligent central control mechanism to dynamically adjust protocol functions to support resilience to the difficult requirements of WSNs, maintain a scalable architecture, and exhibit improved shows network management and operations in terms of performance and resource utilization. They feature CORALSDN protocol manipulations in real testbeds, providing an environment suitable for hands-on experiments, highlighting the improvements SDN brings to the IoT. The results show that a software-defined method can provide a robust solution for improving network control of IoT networks.

2.10 Securing Internet of Things System Using Software Defined Network Based Architecture

In IoT systems, these low-power devices are a protection challenge for IoT systems because they are highly vulnerable to cyberattacks, which reduces the stability of the system. Software-Defined Networks (SDN) aims to greatly facilitate policy enforcement and dynamic network reconfiguration. This white paper describes several architectures for improving network and system security of IoT integration via SDN [28]. A secure location can be based on a variety of reasons, such as application and network topology. Thus, according to the author he provides a multiple IoT-based SDN architecture to improve network security. The main feature of SDN is that the SDN controller is flexible and allows you to see the behavior of the entire network. There are more advanced methods for the security of IoT SDN systems such as network slices, but they have not yet been thoroughly explored. Therefore, including SDN in your existing IoT platform can not only provide the efficiency of data storage and analysis, but also improve security.

2.11 The Method of Controlling Traffic Paths in IoT-Based Software Defined Network

The authors provide a network management method applicable to the configuration of traffic routes and the functioning of nodes included in software-defined network routes. In other words, the advanced functions of each switch node can be utilized as network resources and as a result, users can provide a traffic path that efficiently supports customized services [29].

Which, If the user can select the data path to the IoT service and configure the configuration (encryption, security, QoS, etc.), it will be convenient for the IoT service to provide efficient service and management. When a user using an IoT terminal requests the setting of a path for a communication service, SDN's central control system provides the user with a list of configurable paths. And because of that proposed method from a list of user-configurable routes, the user can choose the most appropriate route for the service as he requires. In addition, the central control plane can provide users with information about the node functions supported by each switch node composing the path in the form of a selectable menu, and as a result, which of each switch node composing the path can provide the user. At least one node function can be selected for at least one service the user wants. Therefore, the quality of service can be improved because the route can be easily configured to suit the user's requirements [29].

2.12 ID-Based SDN for the Internet of Things

Based on [30] IoT, big data not only brings convenience to people's daily lives, but also increases the potential attack surface of cybercriminals. At the same time, given the resource constraints and heterogeneity characteristics, existing network security solutions can be difficult to achieve ideal results in an IoT environment, a challenge facing IoT big data security. In this case, the benefits introduced by software defined networking (SDN) may solve the problem of IoT security risks. For this reason, the author proposes an identity-based SDN security network architecture called IBSDN. Unlike existing SDN solutions, IBSDN is dedicated to providing reliable and intrinsic services to the IoT from the network side, which includes non-forgery terminal IDs in the data stream. This network-level reliable service prevents IoT devices from consuming limited resources for security and provides greater scalability and manageability for monitoring network security. Firstly, they analyze the characteristics of IoT big data and potential security attacks. Then proposed the following identity-based SDN security architecture: This architecture uses the powerful representation of IPv6 addresses while including terminal identity information. Original address verification technology ensures a trusted and traceable device identity in the management domain. Finally, the security capabilities of the IBSDN architecture are analyzed and key future challenges and opportunities are discussed.

Packet Forwarding Priority (PFP) has been available in off-the-shelf routers for a long time, and support is provided by various models from popular brands such as Cisco and Juniper Networks. Network operators have come to rely on these mechanisms to manage their networks. For example, as a way to rate-limit certain classes of applications (peer-to-peer), PFPs can have a significant impact on the performance of applications beyond what administrators target. PFP can also seriously affect the accuracy of the output of measurement tools and the effectiveness of network troubleshooting procedures [31].

2.13 Modified MAC for Priority Traffic with Slow Decrease of Contention Window and Reservation Based Packet Forwarding in IEEE 802.11 for QoS Provisioning

The authors demonstrate a simple method for enhancing multimedia real-time performance in 802.11 WLANs [32] by implementing Quality of Service Manager (QoSM) for distinguishing services with two queues on an 802.11 media access controller. Contention window (SD) and reservation-based packet transmissions are slowly decreasing. The proposed scheme improves the performance of multimedia real-time service in infrastructure-based WLAN by coexistence of non-real-time traffic, verified with the help of ns-2. They consider a software upgrade-based deployment method to provide limited QoS for real-time multimedia service extension via MAC controller in 802.11 WLAN. The scheme needs to scale further to support voice traffic.

2.14 Integrated Services to Differentiated Services Packet Forwarding: Guaranteed Service to Expedited Forwarding PHB

Authors id this paper [33], outlines the design and simulation of packet transport algorithms between the Integrated Services network domain and the Differentiated Services network domain. The algorithm for sending packets in guaranteed class of service flows first to the transport PHB. This algorithm is assumed to be implemented in a pipeline processor configuration that separates the input packet processing function and the output packet processing function. The remaining jitter values are used to prioritize packet transmission. Measures of throughput and packet loss rates are shown. From simulations, they have found that this algorithm can yield good packet throughput and acceptable packet loss.

3 Problem Statement

In IoT applications for smart cities, a large number of shot messages need to be sent from a source node to a destination node with high level of network performance metrics such as the power efficiency, the transmission time, the delay etc. One of the main challenges of IoT in smart cities is how to schedule the access to the communication channel while the network nodes become inactive most of the time and regularly send their packets when needed without human interaction. However, some urgent packets in smart city applications such as informing a hospital about emergency cases occurred in a specific location needs fast packets delivery to the destination node, and giving this prioritize packet all permission in order to ensure high transmission ratio with low latency. To address this issue in big data smart cities, many scheduling algorithms have been proposed recently for big data smart cities. All of these algorithms focus on how to ensure high scheduling network efficiency and appropriate allocation of network resources. For example, scheduling algorithms such as Earliest deadline First (EDF) [34], or First Come First Served (FCFS) [35], use data packets for differentiation between priority and nonpriority packets [36], while other studies proposed algorithms that give the priority packets a higher permission for single node transmission that preempt packets with low priority [37]. Quality of Service (QoS) approaches have been used extensively as scheduling methods in big data smart cities. In QoS approaches, three factors affect the priority packets transmission which are the type the transmission node (i.e., the source node), the type of data and the timeline constrains for transmitted packets which cause difficulties in traffic management for large packets transmission environment such as big data smart cities. In our paper, we try to simplify the class-based packets transmission in big data smart cities. In other words, we proposed high performance calss-based priority packets scheduling mechanism for emergency application of big data smart cities. The main contributions of this paper are the following:

- The proposed scheduling algorithm is class-based algorithm which enable the destination node to explore information about the source node data packets. The priority packets will be transmitted first based on a threshold factor for the packet deadline. We call our proposed mechanism as Priority Packet Deadline First (PPDF). Packets that their deadline are expired will be dropped from the network channels in order to achieve high efficiency of network resources and low transmission overhead. In smart cities, depending on a packet deadline threshold instead of the threshold of packet transition time will lead to better utilization on the network bandwidth.
- We take the advantages of using centralized communication scenario based on SDN controller to handle all the required transmission for priority packets to the final destination node. The main idea behind the SDN controller is SDN to enhance the delivery of priority

packets and ignoring the nonpriority packets (reduce the number of packets transmitted to the destination node). In addition, SDN controller can reduce the power consumption for IoT nodes and hence enhancing the smart cities network lifetime.

- In the network module for the proposed PPDF as well as SDN techniques, we proposed three algorithms, that are the broadcasting data packets algorithm at source nodes, determining priority packets algorithm at SDN controllers, and scheduling/forwarding priority packets algorithm at SDN controllers.
- We compare our proposed PPDF scheduling algorithm with existing scheduling schemes used in big data smart cities in order to demonstrate how the proposed PPDF algorithm achieve high performance metrics in terms of average waiting time, priority packets loss rate, priority packets end-to-end delay and the average energy consumption.

4 Priority Packet Deadline First (PPDF) with SDN Controller

In the proposed PPDF algorithm, static nodes are distributed randomly in the sensing area. We have source nodes, intermediate nodes and master nodes (the SDN controllers). Three different types of data packets are generated by source nodes. First, high priority data packets, which are the packets need to be sent correctly (quickly) to the SDN controllers. These data packets preempt both medium priority data packets and nonpriority data packets. High priority data packets (level 1) such as packets that contains data about fire or emergency situations in a specific location. Second, medium priority data packets (level 2), which are the packets that can preempt the nonpriority data packets. These data packets need to be sent to the SDN controller in order to carry information about the network capabilities in the presence of the fire or the emergency location. Third, nonpriority data packets (level 3) which are the packets that have very long deadline, and they can be delivered to the SDN controller or not depending on the heavy traffics of the network.

Named Data Networking (NDN) in IoT can use the main feature of broadcast scenario of data packets to the SDN controllers where the source nodes broadcast data packets to all their neighbors. These neighbor nodes or intermediate nodes forward data packets to the SDN controllers or provide them from their cash memories. Thus, the probability of correctly delivering the data packets based on broadcast scenario to the SDN controllers increased dramatically. Redundant data packets with different priority levels (i.e., level 1, 2 and 3) can be delivered to the SDN controllers, which take the responsibilities for data packets scheduling, and the access the Medium Access Control (MAC) channel in order to forward the data packets to the final destination nodes.

4.1 The Network Model of PPDF Algorithm

The network model of the proposed PPDF algorithm with SDN controller in shown in Fig. 1. The green nodes represent the source nodes that detect the emergencies and send their level 1 data packets through the yellow intermediate nodes that broadcast the level 1 data packets to the red SDN controllers. The orange source nodes are the nodes that send the level 2 and level 3 data packets. Here, the intermediate nodes forward (broadcast) the receiving data packets in their broadcast ranges, so the redundant data packets are delivered correctly to the SDN controllers (red nodes) in IoT environment. The SDN controllers organize the access to the control channel where the level 1 data packets with short deadline will be sent first to the destination nodes (blue nodes). The destination nodes represent the final destinations for the data packets delivery such as the hospitals or the police stations.

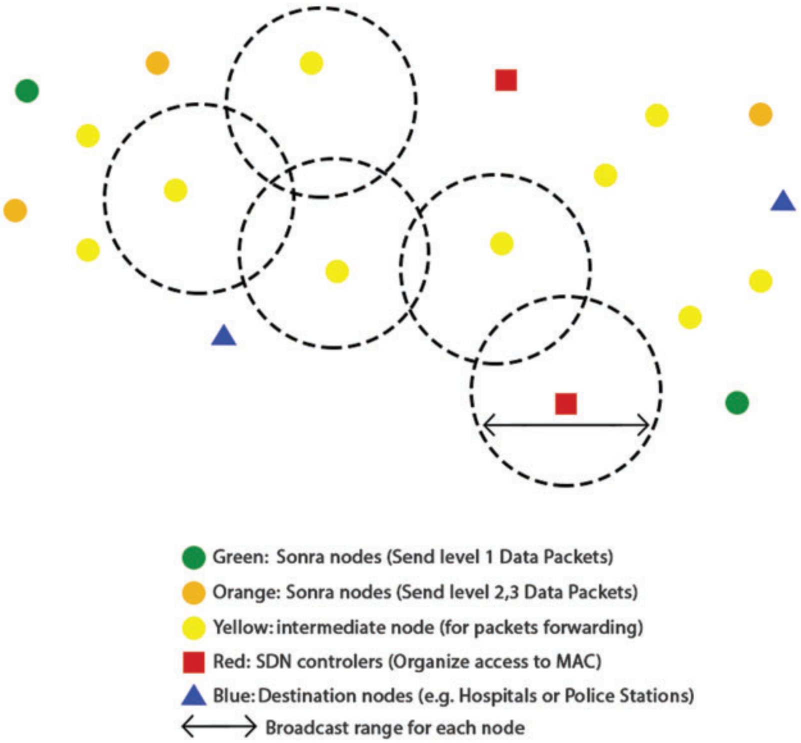

● Green: Sonra nodes (Send level 1 Data Packets)
● Orange: Sonra nodes (Send level 2,3 Data Packets)
● Yellow: intermediate node (for packets forwarding)
■ Red: SDN controlers (Organize access to MAC)
▲ Blue: Destination nodes (e.g. Hospitals or Police Stations)
⟷ Broadcast range for each node

Figure 1: Network model for the proposed PPDF algorithm with SDN controllers

All nodes in the proposed network model have specific tables, and data packets are distributed into slots as shown in Fig. 2. In this figure, the source nodes have the source structure which includes the Content of Data Packets (CDT), the Waiting Table (WT), the Forwarding Table (FT), and the Neighborhood Table (NhT), while the SDN controllers utilize controllers' structure which have the Network Table (NT) that contains the whole network information. This networks table includes the Routing Table (RT) to the destination nodes based on the shortest path algorithm (i.e., Dijkstra algorithm) and the Forwarding Table (FT_SDN). The data packets are divided into slots which contains information such as node ID number, SDN controller ID number, neighbor node ID number, content name, the deadline timer (i.e., our proposed threshold to distinguish between the three levels of data packets) and the energy threshold.

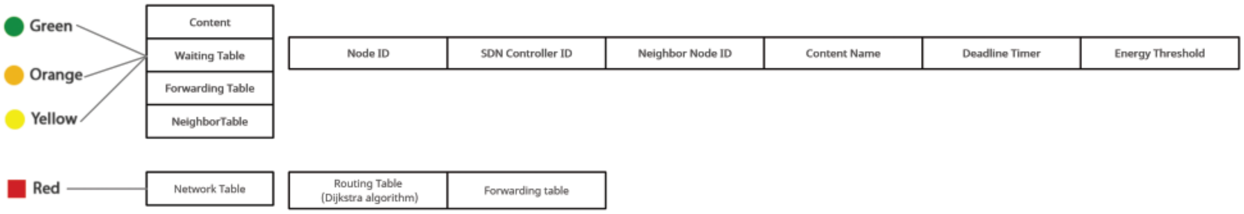


Figure 2: Nodes' structure and data packet frames

4.2 PPDF Algorithm Architecture

Data packets will be broadcasted from the source nodes to the SDN controllers through the intermediate nodes in the proposed PPDF algorithm. The PPDF algorithm has four main architectures, which are explained in the following subsections. These four architectures include connecting source nodes with SDN controllers, access control mechanism, filtering data packets and forwarding data packets (see in Fig. 3.

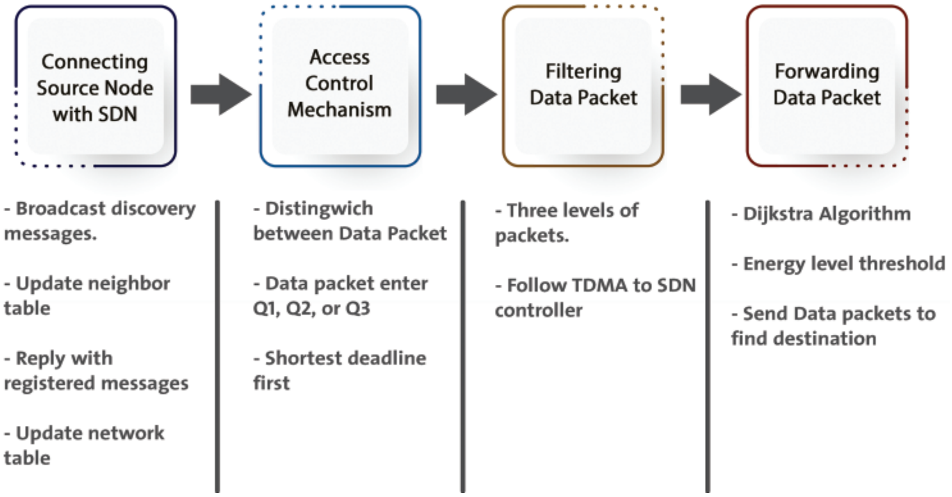

Connecting Source Node with SDN
Access Control Mechanism
Filtering Data Packet
Forwarding Data Packet
- Broadcast discovery messages.
- Update neighbor table
- Reply with registered messages
- Update network table
- Distingwich between Data Packet
- Data packet enter Q1, Q2, or Q3
- Shortest deadline first
- Three levels of packets.
- Follow TDMA to SDN controller
- Dijkstra Algorithm
- Energy level threshold
- Send Data packets to find destination

Figure 3: Four architectures of the proposed PPDF algorithm

4.2.1 Connecting Source Nodes with SDN Controllers

The SDN controller sends its location information via broadcast messages (Discover_Packet) to all Source Nodes (SN) in the network. At SN, when the broadcast messages arrive correctly, the SN checks if the locations information of SDN controller didn't store before in the Neighborhood Table (NhT), then the SN will update its own NhT and connect to the registered SDN (i.e., the SDN controller ID number). The information that is updated in NhT include node ID number, SDN controller ID number, neighbor node ID number, content name, and the deadline timer. After that, the SC sends broadcast message (Registered_Packet) toward registered SDN controller. Therefore, intended SDN controller will pick up the Registered_Packet for further processing. Once the exact SDN controller receives the Registered_Packet, the Network Table (NT) which includes information about the whole network nodes will be updated at the SDN controller (i.e., see in Algorithm 1).

Now, the connection between the SDN controllers and the SN is discovered by Algorithm 1. The SDN controller will take the responsibility to manage data packet transmission to the final destinations. Once the data packets (i.e., packets indicate emergency situations that are occurred in specific locations) arrive to the SDN controllers, they will be processed by entering three main stages, which are the filtering data packets access control, and forwarding data packets.

Algorithm 1: Broadcasting Data Packets Algorithm at Source Nodes to Build Connection

```
Input: Source Nodes (S), SDN Controllers (SC)
Output: Update Network Table (NT) at SDN Controllers
SC.SendBroadcast (Discover_Packet)
if   BP_Packet.NotStored.in.S( )
     Connect_S_To_SC ( )
     Update_ NhT ( )
     S.SendBroadcast (Registered_Packet)
     do
     SC_Update_NT ( )
     while (Registered_Packet received by SC)
else
     S.Drop (Discover_Packet)
end if
```

4.2.2 Access Control Mechanism

In the access control mechanism, the source node distinguishes between data packets based on their types. As mentioned earlier, there are three different types of data packets with various priorities (i.e., level 1, 2 and 3 priority data packets). Once the source node has data packet needs to be sent to the SDN controller and the conditional access control will check the frames of the data packet. The deadline timer in the packet frame will be checked in order to compare the priorities of various data packets of the source node or the neighbor nodes which are located in its range. Every source node can send data packets and access to the control channel through three different types of access channels. Therefore, queue 1 ($Q1$) will handle the level 1 data packets that have the highest priorities, queue 2 ($Q2$) will handle the level 2 data packets that have the medium priorities, and queue 3 ($Q3$) will handle the level 3 data packets that have the lowest priorities.

Thus, data packets will be stored in the control channel based on the deadline timer (i.e., data packets with shortest deadline first access to the control channel). In Fig. 4, two source nodes have different data packets that have various priority levels. The figure shows how the data packets with shortest deadline enter the access channel and located in the front of the queue for two source nodes located in the same transmission range.

4.2.3 Filtering Data Packets

In filtering data packets architecture, data packets need to be sent to the final destination must be compared based on their deadline which is the threshold that determine which packets can access to the communication channel. In this proposed PPDF algorithm, there are two scenarios for the data packets that compete to access two the communication channel (see in Algorithm 2).

First scenario, multiple source nodes have multiple data packets compete to access to the communication channel which have the same priority. In filtering data packets architecture, suppose the packets priorities are denoted as $p_{r1} = p_{r2} = p_{r3} = \dots p_{rk}$; where p_{r1} is the priority for the first data packet needs to be accessed to the communication channel. Then, $deadl_1$, $deadl_2$, $deadl_3$ $\dots deadl_k$ must be compared; $deadl_1$ is the deadline timer for the first data packet need to be accessed to the communication channel. If the node x has a data packet with deadline timer which is lower than the threshold level t, then this packet is level 1 emergency packet.. if $deadl_1 = deadl_2 = deadl_3 = \dots deadl_k$, then the control channel will pick up any level 1 data packet from nodes 1, 2, 3, or k to access to the communication channel because all of their packets are level 1 data packets with the same priority.

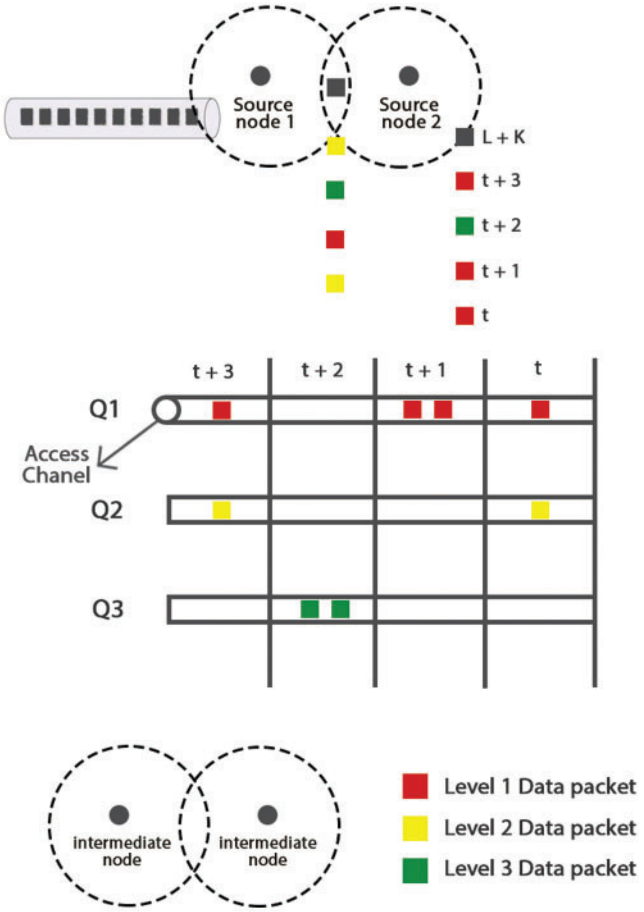


Figure 4: Two source nodes in the same transmission range access to the control channel and how to priorities the data packet in the channel queue

Second scenario: multiple source nodes have multiple data packets compete to access to the communication channel which have different priorities. Suppose that as $p_{r1} < p_{r2} < p_{r3} < \ldots p_{rk}$, that is the data packets for the node 1 has lower priority than the data packet of node 2. Here, we compare the deadline for these data packets where $deadl_1 < deadl_2 < deadl_3 < \ldots deadl_k$, then the packet which has the lowest deadline timer among nodes will be picked to be entered in the front of the communication channel. (i.e., $S_1 < t$ compared with $S_k < t$; where S is the source node and t is the threshold level).

Now, either scenario filters the data packet need to be entered the communication channel. Time Division Multiple Access (TDMA) MAC protocol are utilized to push the filtered data packets into the communication channel in order to deliver these priority packets to the SDN controller for further processing. Once the SDN controllers receive the data packets, they will follow the shortest path algorithm in order to deliver the data packets to the final destinations as we will explained in the forwarding subsection (i.e., Subsection 4.2.4).

Algorithm 2: Filtering Data Packets

Input: Connected Source Nodes (S) with SDN Controllers (SC)
Output: Filtering Data Packets based on Priorities

```
1.  S.Send (Data_Packet)
2.  if   (Data_Packet == 1)
3.      call.First_Scenario ( )
4.      call.TDMA (Data_Packet)
5.      SC.Receive (Data_Packet)
6.  else
7.      call.Second_Scenario ( )
8.      call.TDMA (Data_Packet)
9.      SC.Receive (Data_Packet)
10. end
11. call.First_Scenario ( )
12.   for $i = 1$ to $k$
13.     if ( $p_{ri} = p_{rk}$ )
14.     check($deadl_i$)
15.     if ($deadl_i = deall_k$)
16.     Randomly select Data_Packet
17.       if ($deadl_i \leq t_1$)
18.       Put (Data_Packet = $Q1$)
19.       else if ($t_1 < deadl_i < t_2$ )
20.       Put (Data_Packet = $Q2$)
21.       else
22.       Put (Data_Packet = $Q3$)
23.       end if
24.     end if
25.     end if
26.   end for
27.   Return_to_line 4
28. call.Second_Scenario ( )
29.   for $i = 1$ to $k$
30.     if ( $p_{ri} \leq p_{rk}$ )
31.     check($deadl_i$)
32.     if ($deadl_i \leq deall_k$)
33.     select Data_Packet $i$
34.       if ($deadl_i \leq t_1$)
35.       Put (Data_Packet = $Q1$)
36.       else if ($t_1 < deadl_i < t_2$ )
37.       Put (Data_Packet = $Q2$)
38.       else
39.       Put (Data_Packet = $Q3$)
40.       end if
41.     end if
```

(Continued)

```
42.     end if
43.   end for
44.   Return_to_line 8
```

4.2.4 Forwarding Data Packets

After completing the connection between the source nodes and the SDN controller, and perpetrating the access control stage with distinguishing between priority and nonpriority data packets, data packets will follow TDMA MAC protocol in order to be delivered to the SDN controllers. Once the data packets arrived to the corresponding SDN controller, the SDN controller will check the NT table for matching with the received data packets. If the match is applicable, then the SDN controller will follow the Dijkstra shortest path algorithm in order to deliver the data packets to the final destination. If the match is inapplicable, then the data packets will be dropped. Hence, the redundancy of data packets will be reduced as well as saved the network resources. In addition, in our PPDF algorithm, we also use the energy threshold level (*El*) (i.e., nodes with remaining energy level which are greater than 20%) in order to determine which nodes can be selected as members in the shortest path between the SDN controllers and the final destinations. We assumed 20% as the threshold level of the remaining energy. However, it can be tuned as required by the network designer (See in Algorithm 3).

Algorithm 3: Forwarding Data Packets

Input: Connected Source Nodes (S) with SDN Controllers (SC)
Output: Forwarding Data Packets to Final Destination

```
SC.Receive (Data_Packet)
  SC.check (NT)
for Data_Packet = 1 to $j$
if (Data_Packet = NT)
    if ($El > 20\%$ )
    SC.Compute_Dijkstra_Algorithm ( )
        if (Dijkstra_calculated)
        SC.send_to_Destination (Data_Packet)
        else
          SC.Drop (Data_Packet)
      else
      SC.Drop (Data_Packet)
    else
    SC.Drop (Data_Packet)
            end if
        end if
    end if
    end for
```

5 Simulation Results and Performance Analysis

In this section, we conduct the simulation results to evaluate our proposed PPDF algorithm in comparison with similar algorithms utilized in big data smart cities of IoT. The proposed PPDF algorithm is compared with Dynamic Multilevel Priority packet scheduling algorithm (DMP) [38], normal NDN flooding mechanism in IoT (NDN_Flooding), and First-Come, First-Served (FCFS) algorithm [35] in terms of average waiting time, packet loss rate, end-to-end delay and average energy consumption. Based on NS2 simulator, we conduct a scenario where priority packets are needed to be sent to the final destination (hospital) in order to request an imbalance for cars accident located in specific location in a geographical area of IoT. Twenty source nodes, which are located in coverage area of the cars' accident, send level 1 (high priority data packets) through the intermediate nodes to be delivered to the closest SDN controller using TDMA MAC protocol. The corresponding SDN controller sends the received priority packets to the hospital which is located far away in the geographical area. The SDN controller follows the proposed PPDF algorithm for the priority data packets transmission. In the simulation, the normal data packets (level 2 and 3) are generated every 10 min randomly, but the level 1 data packets are sent once the cars accidents are occurred in the sensing area. The simulation settings are demonstrated in Tab. 1.

Table 1: Simulation parameters

Parameter	Value	Parameter	Value
Simulation time (s)	5000 s	Transmission speed (kb/s)	300 kb/s
Field size (m)	500 m × 500 m	Initial energy (joule)	50 joule
Number of source nodes	1000	Energy threshold	20%
Number of SDN controllers	50	Transmission power (dbm)	20 dbm
Transmission rage (m)	30 m	Wireless interface	IEEE 802.15.4
Packet size (byte)	200, 400, 600, 800 and 1000	Propagation	Two ray ground model

For the first performance metric (i.e., the average waiting time), the simulation results show that when generating various data packets from source nodes with different transmission rate, the network conditions can ensure different data packets transmission to the SDN controllers and then to the final destination node without collisions that consume network resources. In other words, the simulation network model can be tuned from sending high priority data packets to low priority data packets easily. Fig. 5 illustrates the average waiting time for the proposed PPDF algorithm against DMP, NDN_flooding and FCFS algorithms. The figure shows how the average waiting time can increase when increasing the packets sizes. The figure also demonstrates that the data packets with priority level 1 (i.e., Fig. 5a) has lower average waiting time than the data packets with priority level 2 and 3 (i.e., Figs. 5a and 5b). Thus, the level 1 data packets are delivered to the destination nodes faster than the level 2 and 3 data packets due to the fact that the competition to access to the communication channel is depended on the shortest deadline first. In addition, the proposed PPDF algorithm outperforms the existing DMP, NDN_flooding, and FCFS algorithms in term of the average waiting time. The normal flooding NDN algorithm has longer waiting time and packets are competed difficulty to access to the communication channel.

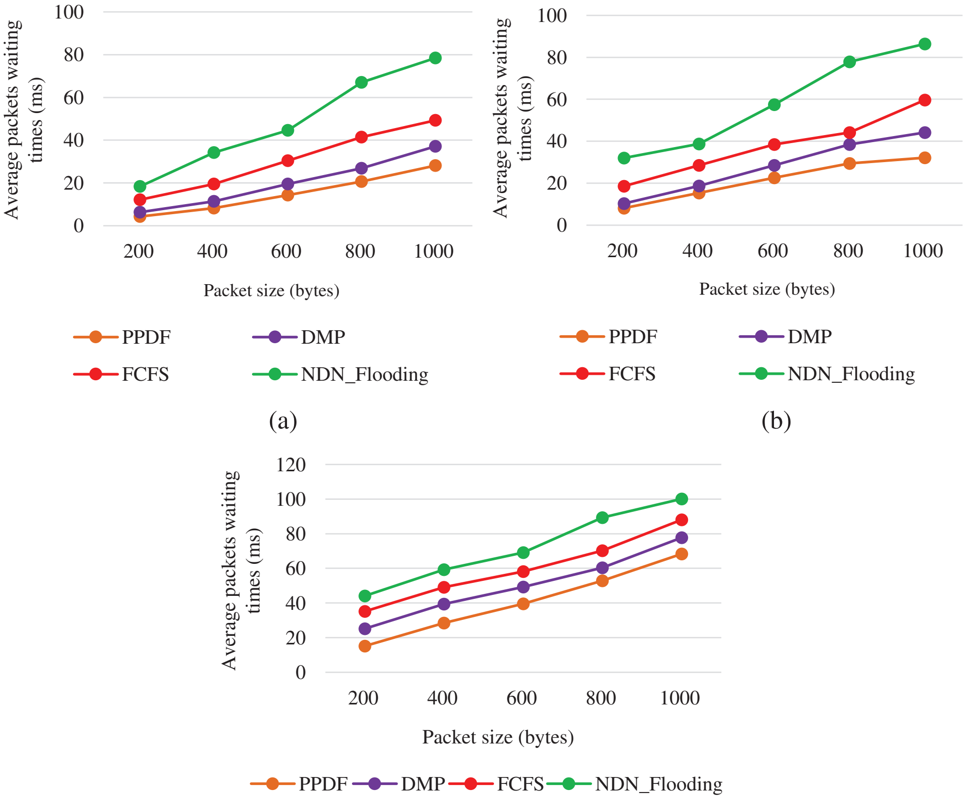
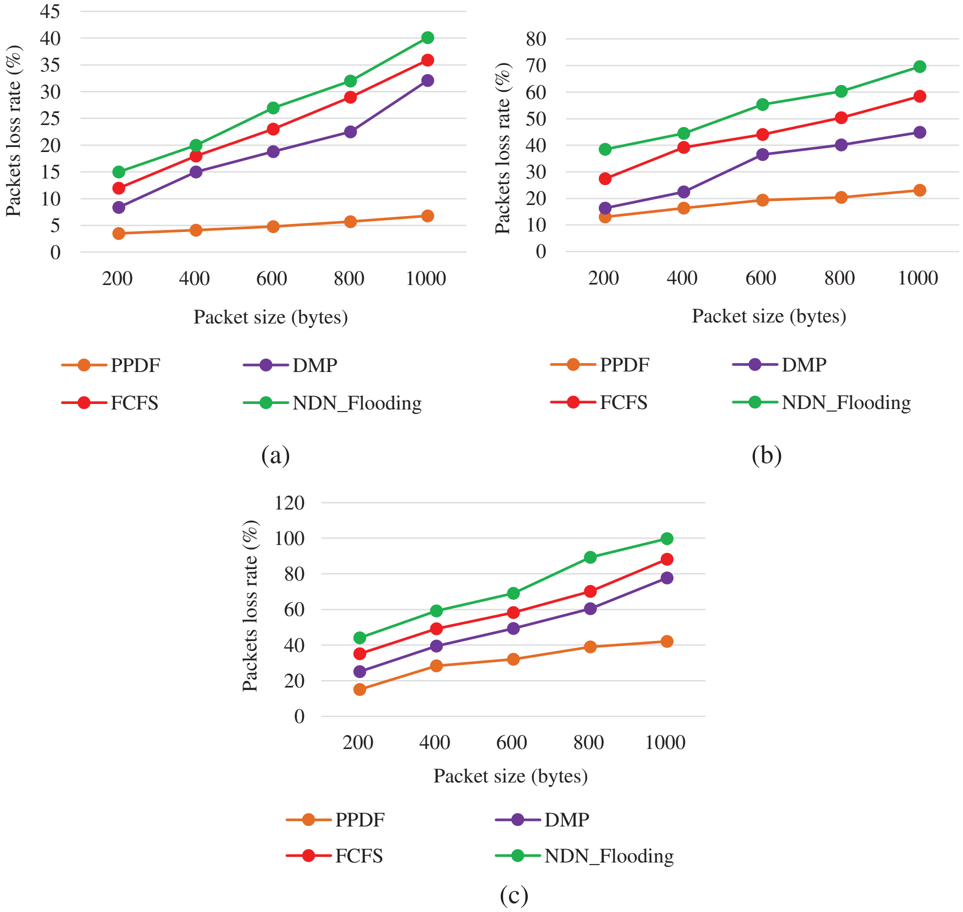


Figure 5: Average waiting time. (a) Level 1 data packets, (b) Level 2 data packets, (c) Level 3 data packets

For the second performance metric (i.e., packets loss rate), Fig. 2a illustrates how the level 1 data packets loss rate increase slowly when increasing the packets sizes. However, in the proposed PPDF, the packet loss rate is not obvious as it can reach 6.80% when the packet length is 1000 byte. It is followed by the DMP and FCFS algorithms while the NDN flooding algorithm has the highest packets loss rate (i.e., around 40.10% when the packet size is 1000 byte). In addition, Figs. 6b and 6c still demonstrate that the proposed PPDF algorithm outperforms the existing algorithms, and has lower packets loss rates for both the level 2 and 3 data packets. Also, increasing the packet lengths causes increasing in the packets loss rates gradually. The network conditions in FCFS algorithm have an affect on the packets retransmission due to the collision possibilities. Packets access to the communication channel suffer from the MAC layer collision which means that the packets will be dropped before they are arrived to the final destinations. The same scenario is occurred in DMP algorithm, that is when the communication channel solves the collision to handle packets transmission, the data packets will be dropped before arriving to the final destinations due to the deadline expiration possibilities. In normal NDN flooding algorithms, packets loss rates are high due to both collision and inefficient MAC protocol that flood the network with heavy traffic without distinguishing between high and low priority packets. Thus, from previous figures, we can conclude that in PPDF algorithm the average waiting time for our proposed algorithm is low in comparison with the existing algorithms, so the packet loss rate will be low too due to that fact that data packets will be arrived to the final destination (the hospital) before the deadline expiration. Even if the data packets with priority level 2 and 3, the proposed PPDF still has high efficiency packets loss rates which make this algorithm is the right choice for scheduling in heavy traffic IoT environment.



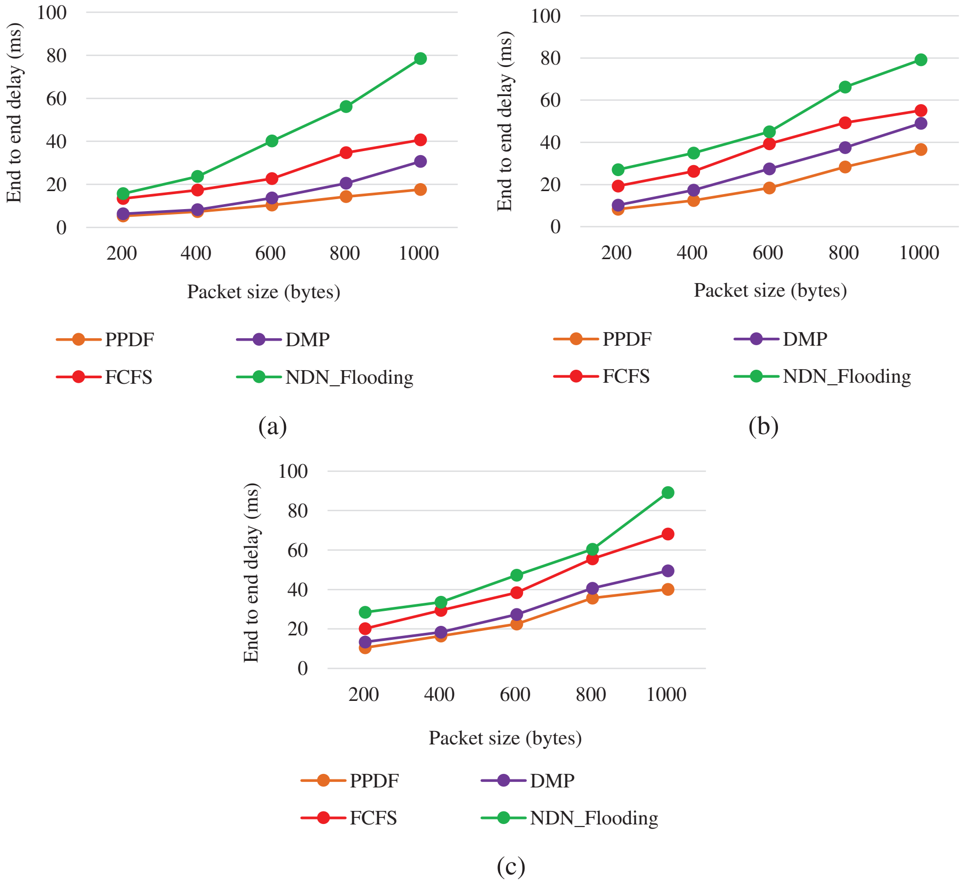
(a)

(b)

(c)

Figure 6: Packets loss rate: (a) Level 1 data packets, (b) Level 2 data packets, (c) Level 3 data packets

For the third performance metric (i.e., end-to-end delay), Fig. 7a illustrates how the end-to-end delay of the proposed PPDF algorithm is low in comparison with the other algorithms. When the packet length is short, the PPDF algorithm has almost the same end-to-end delay of the DMP algorithm. Around 5.3 ms is the end-to-end delay for the proposed PPDF when the packet length is 200 bytes. This delay increased to 17.6 when the packet size is 1000 bytes. The DPM algorithm has 6.2 ms end-to-end delay for when the packet length is 200 bytes, while it has 30.6 end-to-end delay when the packet size is 1000 bytes. In DPM algorithm, the TDMA is the MAC protocol that seems suitable for scheduling access to the communication channel when the packet length is short. However, when increasing the packet size, the time frame for packets scheduling will be increased too resulting in increasing the end-to-end delay. Figs. 7b and 7c show that the proposed PPDF algorithm still allocate network resources for level 2 and 3 data packets.



(a)

(b)

(c)

Figure 7: End-to-end delay. (a) Level 1 data packets, (b) Level 2 data packets, (c) Level 3 data packets

For the fourth performance metric (i.e., energy consumption), Fig. 8 illustrates how the energy level for the proposed PPDF outperform the existing algorithms due to the fact that every node has its own energy threshold that control the selection of intermediate nodes which involve in the transmission processes. The figure shows that the energy consumption will be decreased when increasing the number of SDN controllers as they considered to be the master nodes with high features for packets transmission. The highest number for SDN controllers in our simulation is 50 controllers. When the SDN controller is 50, the energy consumption for the proposed PPDF algorithm is 4 joules, while the energy consumption for the DMP algorithm, FCFS algorithm and NDN_flooding algorithm are 6, 10 and 22 joules, respectively. Thus, the highest energy preserving algorithm is the proposed PPDF algorithm followed by DMP algorithm, FCFS algorithm and NDN_flooding algorithm respectively.

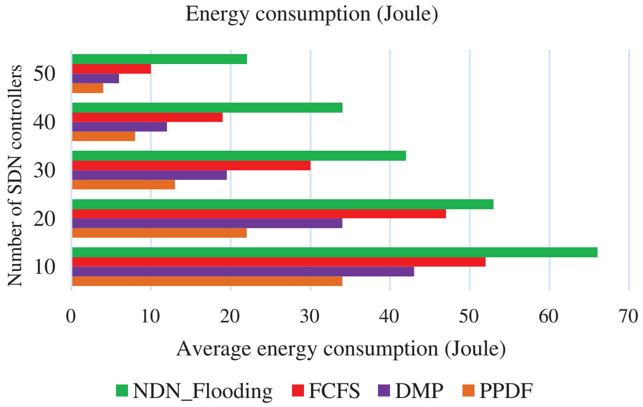


Figure 8: Energy consumption (joules)

6 Conclusion

In this paper, the researchers proposed PPDF scheduling algorithm for emergency applications of big data smart cities. The proposed PPDF preserves network resources via transferring the priority information between source nodes and destinations. Source nodes with high priority data packets, that represent emergency situations indications, will be delivered through high priority communication channels, while packets with low priority information will be delivered through low priority data channels. Nodes compete to access to the communication channel by using TDMA MAC protocol in the transmission path between the source nodes and the SDN controllers. However, packets follow the earliest deadline first in the transmission path between the SDN controllers and the destination nodes. In our paper, the authors proposed four architectures for the PPDF algorithm which are connecting the source nodes with SDN controllers, the access control mechanism, determining priority packets stage and forwarding process. The proposed PPDF algorithm is compared with existing well known algorithms used in big data smart cities, and the simulation results shows how our proposed algorithm has high performance efficiency in terms of the average waiting time, the packet loss rate, the end-to-end delay and the energy consumption. In the future work, we will increase the flexibility of the PPDF algorithm via increasing the number of static nodes and adding mobile nodes in the network model. More simulation performance metrics can be measured with mobile nodes which are moving from one location to another to be closed to the SDN controllers.

Acknowledgement: We deeply acknowledge Taif University for supporting this study through Taif University Researchers Supporting Project Number (TURSP-2020/150), Taif University, Taif, Saudi Arabia.

Funding Statement: This study is supported through Taif University Researchers Supporting Project Number (TURSP-2020/150), Taif University, Taif, Saudi Arabia.

Conflicts of Interest: The author declare that he has no conflicts of interest to report regarding the present study.

References

[1] P. Ta-Shma, A. Akbar, G. Gerson-Golan, G. Hadash, F. Carrez *et al.,* "An ingestion and analytics architecture for IoT applied to smart city use cases," in *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 765–774, 2018.

[2] Q. Cui, Y. Wang, K. Chen, W. Ni, X. Tao *et al.,* "Big data analytics and network calculus enabling intelligent management of autonomous vehicles in a smart city," in *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2021–2034, 2019.

[3] L. Ang, K. P. Seng, A. M. Zungeru and G. K. Ijemaru, "Big sensor data systems for smart cities," in *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1259–1271, 2017.

[4] P. Kulkarni and K. B. Akhilesh, "Big data analytics as an enabler in smart governance for the future smart cities," in *Smart Technologies*, Springer, Singapore, pp. 53–65, 2020.

[5] Y. Wang, P. Tosh, J. Lee, Y. Lee, S. Henderson *et al.,* "An interdisciplinary educational project connecting smart city technology with local communities," in *IEEE Int. Smart Cities Conf. (ISC2)*, Kansas City, MO, USA, pp. 1–2, 2018.

[6] G. R. Ceballos and V. M. Larios, "A model to promote citizen driven government in a smart city: Use case at GDL smart city," in *IEEE Int. Smart Cities Conf. (ISC2)*, Trento, Italy, pp. 1–6, 2016.

[7] A. Founoun and A. Hayar, "Evaluation of the concept of the smart city through local regulation and the importance of local initiative," in *IEEE Int. Smart Cities Conf. (ISC2)*, Kansas City, MO, USA, pp. 1–6, 2018.

[8] E. Mardacany, "Smart cities characteristics: Importance of buit environments components," in *IET Conf. on Future Intelligent Cities*, London, pp. 1–6, 2014.

[9] S. Chaudhary, R. Johari, R. Bhatia, K. Gupta and A. Bhatnagar, "CRAIoT: Concept, review and application(s) of IoT," in *4th Int. Conf. on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, Ghaziabad, India, pp. 1–4, 2019.

[10] E. P. Yadav, E. A. Mittal and H. Yadav, "IoT: Challenges and issues in Indian perspective," in *3rd Int. Conf. on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, Bhimtal, India, pp. 1–5, 2018.

[11] W. Hu, W. Sun, Y. Jin, W. Guo and S. Xiao, "An efficient transportation architecture for big data movement," in *9th Int. Conf. on Information, Communications & Signal Processing*, Tainan, Taiwan, pp. 1–5, 2013.

[12] R. Li and H. Asaeda, "Secure in-network big data provision with suspension chain model," in *IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS)*, Honolulu, HI, USA, pp. 825–830, 2018.

[13] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher and B. Ohlman, "A survey of information-centric networking," in *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, 2012.

[14] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs *et al.,* "Networking named content," in *The 5th Int. Conf. on Emerging Networking Experiments and Technologies (CoNEXT'09)*, Rome, Italy, pp. 1–12, 2009.

[15] W. Sun, F. Li, W. Guo, Y. Jin and W. Hu, "Store, schedule and switch-A new data delivery model in the big data era," in *15th Int. Conf. on Transparent Optical Networks (ICTON)*, Cartagena, Spain, pp. 1–4, 2013.

[16] V. Kakkar, "Scheduling techniques for operating systems for medical and IoT devices: A review," in *Global Journal of Computer Science and Technology: AHardware & Computation*, vol. 17, USA, no. 1, pp. 3–19, 2012.

[17] B. Baranidharan and K. Saravanan, "ETSI: Efficient task scheduling in internet of things," in *International Journal of Pure and Applied Mathematics*, vol. 117, India, no. 22, pp. 229–233, 2017.

[18] K. Kavitha and G. Suseendran, "Priority based adaptive scheduling algorithm for IoT sensor systems," in *Int. Conf. on, Automation, Computational and Technology Management (ICACTM)*, London, UK, pp. 361–366, 2019.

[19] I. Avgouleas, N. Pappas and V. Angelakis, "Scheduling services on an IoT device under time-weighted pricing," in *IEEE 28th Annual Int. Symp. on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Montreal, QC, Canada, pp. 1–5, 2017.

[20] N. H. Bui, C. Pham, K. K. Nguyen and M. Cheriet, “Energy efficient scheduling for networked IoT device software update,” in *15th Int. Conf. on Network and Service Management (CNSM)*, Halifax, NS, Canada, pp. 1–5, 2019.

[21] A. Prasanth, J. A. George and P. Surendram, “Optimal resource and task scheduling for IoT,” in *Int. Conf. on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, Sakhier, Bahrain, pp. 1–4, 2019.

[22] Y. Ting-dong and P. Xuan-rong, “Research on scheduling method for collaborative detection traffic in the bottom of IOT,” in *Chinese Automation Congress (CAC)*, Jinan, China, pp. 7699–7703, 2017.

[23] S. El Mendili, Y. El Idrissi and N. Hmina, “Big data processing platform for smart city,” in *Int. Symp. on Advanced Electrical and Communication Technologies (ISAECT)*, Rabat, Morocco, pp. 1–7, 2018.

[24] F. Al-Turjman, L. Mostarda, E. Ever, A. Darwish and N. Shekh Khalil, “Network experience scheduling and routing approach for big data transmission in the internet of things,” in *IEEE Access*, vol. 7, pp. 14501–14512, 2019.

[25] A. Murk, W. Malik, I. Mahmood, N. Ahmed and Z. Anwar, “Big data in motion: A vehicle-assisted urban computing framework for smart cities,” in *IEEE Access*, vol. 7, pp. 55951–55965, 2019.

[26] A. Al Hayajneh, M. Bhuiyan and I. McAndrew, “Improving internet of things (IoT) security with software-defined networking (SDN),” in *Computers*, vol. 9, no. 1, pp. 1–14, 2020.

[27] T. Theodorou and L. Mamatas, “CORAL-SDN: A software-defined networking solution for the internet of things,” in *IEEE Conf. on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Berlin, Germany, pp. 1–2, 2017.

[28] S. H. S. Ariffin, “Securing internet of things system using software defined network based architecture,” in *IEEE Int. RF and Microwave Conf. (RFM)*, Kuala Lumpur, Malaysia, pp. 1–5, 2020.

[29] E. J. Kim, J. A. Jun and N. Kim, “The method of controlling traffic paths in IoT-based software defined network,” in *IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conf. (UEMCON)*, New York, NY, USA, pp. 1–4, 2016.

[30] X. Wang, K. Xu, W. Chen, Q. Li, M. Shen *et al.,* “ID-based SDN for the internet of things,” in *IEEE Network*, vol. 34, no. 4, pp. 76–83, 2020.

[31] G. Lu, Y. Chen, S. Birrer, F. E. Bustamante and X. Li, “POPI: A user-level tool for inferring router packet forwarding priority,” in *IEEE/ACM Transactions on Networking*, IEEE, vol. 18, no. 1, pp. 1–14, 2010.

[32] D. K. Puthal, B. Sahoo, A. K. Turuk and M. Nandi, “Modified MAC for priority traffic with slow decrease of contention window and reservation based packet forwarding in IEEE 802.11 for QoS provisioning,” in *Int. Conf. on Information Technology*, Bhubaneswar, India, pp. 195–196, 2008.

[33] B. Budiardjo, B. Nazief and D. Hartanto, “Integrated services to differentiated services packet forwarding: guaranteed service to expedited forwarding PHB,” in *Proc. 25th Annual IEEE Conf. on Local Computer Networks*, LCN, Tampa, FL, USA, pp. 324–325, 2000.

[34] L. Chen and K. D. Kang, “A framework for real-time information derivation from big sensor data,” in *Proc. 17th Int. Conf. on High Performance Computing and Communications (HPCC)*, New York, NY, USA, pp. 1020–1026, 2015.

[35] Y. Dong, Z. Chen, S. Liu, P. Fan and K. B. Letaief, “Age-upon-decisions minimizing scheduling in internet of things: To be random or to be deterministic?,” *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1081–1097, 2020.

[36] T. Qiu, K. Zheng, M. Han, C. L. Chen and M. Xu, “Dynamic multilevel priority packet scheduling scheme for wireless sensor network,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2042–2051, 2018.

[37] N. Nidal, L. Karim and T. Taleb, “Dynamic multilevel priority packet scheduling scheme for wireless sensor network,” *IEEE Trans. Wireless Communication*, vol. 12, no. 4, pp. 1448–1459, 2013.

[38] O. Chipara, C. Lu and G. -C. Roman, “Real-time query scheduling for wireless sensor networks,” in *Proc. Real-Time Systems Symposium*, Tucson, Arizona, USA, pp. 389–399, 2007.