Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2022) 30: 1439 – 1459

doi:10.55730/1300-0632.3858

Priority enabled content based forwarding in fog computing via SDN

Yasin İnağ[1,*], **Metehan Güzel**[2], **Feyza Yıldırım Okay**[1], **Mehmet Demirci**[1], **Suat Özdemir**[3]

[1]Department of Computer Engineering, Faculty of Engineering, Gazi University, Ankara, Turkey
[2]Department of Computer Engineering, Graduate School of Natural and Applied Sciences, Gazi University, Ankara, Turkey
[3]Department of Computer Engineering, Faculty of Engineering, Hacettepe University,Ankara, Turkey

Received: 04.10.2021 • **Accepted/Published Online:** 25.12.2021 • **Final Version:** 31.05.2022

Abstract: As the number of Internet of Things (IoT) applications increases, an efficient transmitting of the data generated by these applications to a centralized cloud server can be a challenging issue. This paper aims to facilitate transmission by utilizing fog computing (FC) and software defined networking (SDN) technologies. To this end, it proposes two novel content based forwarding (CBF) models for IoT networks. The first model takes advantage of FC to reduce transmission and computational delay. Based on the first model, the second model makes use of the prioritization concept to address the timely delivery of critical data while ensuring the data rate and delay requirements. Extensive simulations are conducted to evaluate the proposed models and uncover their impact on throughput, delay, and loss rate metrics. According to the results, the proposed models ensure efficient transmission and low computational delay. In addition, the second model has the ability to transmit critical data more effectively.

Key words: IoT, fog computing, content based forwarding, priority

1. Introduction

Fog computing (FC) is a recent paradigm that offers additional capabilities to cloud computing by moving the computing, storing, and networking processes to the edge of the network. It is a complementary technology that supports latency-sensitive and location-aware applications by distributing its applications and services to the closer to the end users [1, 2]. Fog servers are located in the middle of the cloud server and end users, and they are preprocess the data before transmitting it to the server in order to alleviate the burden on the cloud server [3].

FC is highly recommended in delay-sensitive applications such as virtual reality and multimedia streaming [4, 5]. Therefore, ensuring quality of service (QoS) during the transmission of data is critical. Furthermore, different flow protocols need to be defined according to the content and importance level of the flowing data since Internet of Things (IoT) data is highly heterogeneous. Instead of transmitting all of the data through the network, aggregating them in the fog servers or prioritizing them according to data types may greatly decrease the density of traffic and increase the performance of the network accordingly [6]. For example, in some cases like healthcare or air quality monitoring, fog servers can be customized for a certain type of service. In such cases, the relevant data generated by geo-distributed IoT devices should be routed over the network to the related fog server that is specialized for that service [7]. To this end, priority-based approaches can be used to

*Correspondence: yasininag@gazi.edu.tr

forward data to the related fog servers. By considering priority during the transmission of data, it is ensured that the important data reaches the related fog server more quickly. It helps fog services ensure sustainability with reduced download delays, less bandwidth usage, improved content availability, and reduced cost [8].

During the transmission of the data from IoT devices to the related fog servers, topology changes may occur due to the large number of devices being connected to the network. Depending on environmental conditions, some active nodes may fail or new nodes may join the network and become active. When instant dynamic changes occur in such a network, managing or controlling the network becomes quite difficult with traditional methods. By separating the network as data plane and control plane, software defined networking (SDN) is offered as a promising solution to this problem with the aim of making the network more flexible and manageable. It provides global control of distributed fog servers, and supports priority management, error control, and energy and delay optimizations [9, 10].

Following the observations given above, in this paper, we tackle the priority based data forwarding problem in IoT networks that employs FC and SDN technologies. We propose two novel content based data forwarding models and implement them using network flow of a realistic IoT based air quality monitoring network. The network flow is generated using the real air quality monitoring data set of İstanbul, which is not only the most populated city in Turkey but also one of the most populated cities in the globe.

In traditional IoT networks, collected data is usually transmitted to a single server without considering the content or application type. Considering the traffic density and bandwidth bottleneck, in the proposed models, there are fog servers each of which is dedicated to a certain data type and/or application. Fog servers are connected to the main cloud server and there are several IoT sensors responsible for generating different types of data. In addition, there is an SDN controller responsible for managing which data should be routed to which fog server. In our first proposed model, called content based forwarding (CBF) via SDN (CBF-SDN), the SDN controller determines the forwarding rules based on IoT data content. The purpose of this approach is to collect same type of data in a single fog server, thereby reducing the computational latency. The second proposed model, named priority enabled content based forwarding via SDN (PCBF-SDN) two types of queue structure are employed when processing IoT data. In the first model, all IoT data are considered equal and sent to their corresponding fog servers. The other queue model prioritizes IoT data based on the application and/or data type. Based on this prioritization, in addition to FC, a copy of critical data is sent to the cloud server directly in case of any emergency situation to make sure critical data is able to be carried effectively and with less delay. The purpose here is to make quick decisions on the cloud server in case of an emergency. In order to present the advantages of the proposed models, extensive simulations are performed using a realistic network flow.

The main contributions of the paper are given as the following:

- To the best of our knowledge, this is the first study that proposes and implements priority enabled content based data forwarding using the SDN technology in FC based IoT networks.
- Two different content based data forwarding models are proposed and implemented to reduce transmission delay.
- Extensive simulations are performed using a realistic air quality monitoring network scenario to show the effectiveness of the proposed models.

The rest of the paper is organized as follows: Section 2 examines the related works on content based forwarding and SDN based prioritization in the literature comprehensively. Section 3 provides brief background information about the building blocks of the proposed models. The details of the proposed models are explained in Section 4. Extensive simulation results are presented and discussed in Section 5. Finally, Section 6 summarizes the paper and presents the future directions to guide the interested readers.

2. Related work

This paper focuses on priority-based content-based forwarding through SDN technology for FC based IoT networks. In SDN-supported networks, resource management is done according to forwarding and data priority. Resource management for SDN-based cloud or SDN-based FC is essential to ensure QoS. The importance of resource management for priority and effective data forwarding has been emphasized in the literature. Creating different flow rules and queue structures increases performance by allocating more resources to priority data [11].

Prioritization of network flows and packets is used to overcome numerous network performance problems. One of the most commonly used and efficient approaches for prioritization is the utilization of SDN. In [12], an SDN based control plane is developed for a university campus that suffers from heavy congestion and low QoS problems. By prioritizing data packets that are identified as important, higher QoS levels are acquired for important services. In the paper, video surveillance is selected as an important service and simulations are implemented on Mininet. In [13], a solution is proposed for transmission disruptions and packet losses happening during dynamic changing of flow rules in SDN controlled networks. The proposed method, namely priority based flow control, provides robust and disruption-free data flows. By prioritizing flow rules, this proposed method reduces the complexity of flow modification process. Therefore, it reduces the time-complexity of control plane processes.

Lu et al. [14] proposed a model that optimizes the use of network resources for emergencies in SDN-supported networks. They provide QoS for process or data forwarding, which takes priority in emergencies. They propose solutions by dividing the network into multiple flow paths according to the priority and essential requirements of the operations. This model provides maximum throughput by calculating different communication paths. Shinkuma et al. [15] proposed machine learning algorithms to prioritize real-time data in SDN-supported networks. Priority is defined by processing the data on edge-cloud servers. Mobile traffic logs are prioritized according to bandwidth usage. Similarly, in [16], a huge volume of data is generated from IoT devices, and high bandwidth is needed to transmit this data to servers. To overcome the mentioned problem, a novel application sensitive model for SDN-supported networks is proposed. While ensuring QoS with the proposed model, delay, jitter, and packet loss metrics are compared.

Diro et al. [17] proposed a new model in SDN-supported networks to provide priority-based service quality. In the paper, different flow paths are created for IoT data with different levels of priority. The flowchart that IoT data will be transmitted over is determined in accordance with the priority of the data. Priority data is transmitted directly, while nonpriority is kept in a separate queue. The proposed model achieves throughput and loss rate advantages by forwarding urgent (priority) packets. Bardalai et al. [18] proposed OpenFlow-based routing methods according to the priority and importance of the data obtained from IoT devices in healthcare systems. According to the authors, the importance of analyzing and transmitting health data after Covid-19 has increased considerably. Therefore, these data must be transmitted without data loss and with minimum delay in the network infrastructure of the priority ones. In the proposed model, queue-based communication is proposed. According to the importance level of the data, it is transmitted over one of the multiple queues. According to the simulation results, throughput and delay have been improved in the transmission of high priority data.

Ghazi et al. [19] simulated a vehicular ad-hoc network (VANET) based model that prioritizes emergency messages and transmits them to the network. Increasing data traffic in networks reveals the importance of priority data forwarding. Furthermore, it is a great advantage that the controller software can be implemented dynamically. For this reason, networks can be reorganized according to changing data priorities and systems. In SDN-supported vehicular networks, priority-based multipath model is recommended to be used in emergency and priority situations. Through the implementation of multipath model, network continuity, resiliency, and priority data transmission are ensured [20, 21].

3. Preliminaries

This section presents the assumptions and brief information about the emerging technologies that are employed in our proposed models.

3.1. Assumptions

In order to define the relationships among layers and realize in-depth analysis, the following assumptions are made.

- Traditional IoT architecture collects and processes all the data coming from different IoT sensors on a single server.
- In the traditional IoT architecture, each sensor conducts IP-based or port-based data transmission.
- In CBF, the data is transmitted to the related fog server which is determined by the SDN controller.
- IoT data can be prioritized based on user defined threshold values. The prioritized data can be differentiated from regular data in fog servers during the data preparation.
- There are multiple paths to the main server having different bandwidths.
- Each sensor is capable of transmitting 900 packets per hour.
- The size of each packet is set to 75 bytes.

3.2. Fog computing

The sharp increase in the number of IoT devices and, accordingly, the massive amount of data produced by these devices push the limits of cloud computing and bring many challenges, including network bottleneck, high latency in response time, and QoS degradation. FC is a promising computing paradigm to compensate for these challenges in cloud computing by getting closer to the network's edge. Thus, compute, storage, and networking services can be performed locally at the edges [22].

FC offers a highly virtualized platform where distributed fog servers are positioned in between cloud servers and end users, which creates a hierarchical structure where fog servers act as relay nodes responsible for connecting them to each other. This structure generally poses three or more tiered structure where each tier has different responsibilities during the transmission of data. The bottom tier called terminal tier consists of smart IoT devices that generate a huge amount of heterogeneous IoT data. Once IoT devices generate its data, terminal tier sends the data to the fog tier located at the middle of the structure. Fog tier includes many distributed fog servers, responsible for storing the data temporarily and preprocessing them before transmission to the cloud tier. Instead of transmitting all the data to the cloud, fog servers generally are utilized to eliminate the redundant data or aggregates [23]. Hence, the bandwidth utilization is reduced, and the load on the cloud server is lightened by reducing the total amount of data. After preprocessing the data in fog tier, the obtained data are sent to the cloud tier for further processing like batch analytics and permanent storage [1, 4].

The application areas of FC are very diversified as from smart cities [24] to smart grid [25], from traffic lights [26] to vehicular systems [27], from healthcare [28] to industrial systems [29]. Especially, due to its geographically distributed structure and proximity to the sources, FC is more applicable for applications that require a fast response time and mobility support such as smart traffic lights, emergence response systems, and augmented reality. FC also supports aggregation, privacy, and security depending on the application type [30].

The connection between fog layers is established through different communication technologies. Due to the proximity to the end users, generally, wireless communication technologies such as wireless local area network, WiFi, 3G, 4G, Zigbee, and even 5G are used for the connection between end users and related fog servers. Also, wired or wireless communications are realized among fog servers. Lastly, fog servers connect to a cloud server over IP core network [4]. According to the status of the devices, the task to be performed, the link quality between devices, and the quality of the requested services, the priority of the communications is determined.

3.3. Software defined networking

In traditional networks, each forwarding element has its own control and data planes. Control plane is responsible for logical operations such as routing optimization, system configuration, and management. On the other hand, data plane is responsible for moving the data to the next destination [31]. In this regard, a network can be considered as a distributed structure consisting of many independent and autonomous devices. After defining a forwarding policy among the network devices, manual configurations may be needed to change the defined policy. For example, in case of an instant topology change, traditional networks fail to adapt to the change due to their limited flexibility and low manageability [32]. In order to compensate for the shortcomings of the traditional networks, SDN technology is proposed [9]. SDN is a rapidly growing technology that enhances IoT networks' abilities of flexibility, programmability, and manageability by decoupling the network as data plane and control plane. This separation makes it possible to move network intelligence to a logically centralized SDN controller that can maintain a global view of the network and provides a programming interface for the network applications [33].

SDN structure consists of three tiers including infrastructure, control and application tiers which are connected to each other through southbound and northbound interfaces. The infrastructure tier is positioned at the bottom of this structure and composed of forwarding elements. According to their flow tables, forwarding elements (i.e. switches, routers, and access points) take actions upon arrival of a packet. The taken action can be transmitting a packet to a controller or an accessing point, dropping or rewriting some headers, etc. The interaction and connection between control and infrastructure tiers are provided by southbound interfaces. OpenFlow [34] is one of the most popular southbound interfaces which enables installing of flow entries in switches, and manages the traffic according to flow entries. Control plane is positioned in the middle tier, containing one or more controller devices that are responsible for controlling and managing all forwarding elements located at the infrastructure tier. Controllers also are able to connect to the application tier which resides at the top of the SDN structure, via northbound interface which provides isolation from the data plane. Hence, applications do not have to be informed about the details of data plane like network topology. The application tier is responsible for programmability, security and network configuration [35].

Since the emergence of the SDN concept, it has been adopted in many domains such as vehicular networks, wireless sensor networks, and cloud computing to simplify network management and provide real-time processing and high availability [36–39]. More recently, the usage of SDN is shifted to the FC paradigm [40]. SDN makes many enhancements like QoS provisioning [41], load balancing [42], task optimization [43], security and reliability [44] possible for dynamic fog based IoT networks. By managing the underlying distributed fog servers and IoT devices, administrative decisions like forwarding optimization, error control, and priority control can be made globally and effectively by SDN controllers.

3.4. Data preparation and application scenario

To show the effectiveness of the proposed models, a realistic application scenario is developed. Using the air quality data provided by the data portal of the Ministry of Environment, Urbanization and Climate Change of Turkey, a data set is created. Among the pollutants monitored by the ministry, nitrogen dioxide (NO2), sulphur dioxide (SO2), particulate matter with size <10 micrometer (PM10), ozone (O3) and carbon monoxide (CO) are selected. The rest of the pollutants are discarded. Using the data set, an air quality prediction scenario is created. The data set is processed and sensor readings are used to acquire a realistic network flow. Then, the network flow is simulated using the proposed models to show the effectiveness of the proposed model. The Ministry of Environment, Urbanization and Climate Change of Turkey which has an air quality monitoring network that consists of 339 air quality monitoring stations spread across the country. Since İstanbul is the most populated city in Turkey and has the highest number of stations, it is selected as the location of this case study. Consequently, as shown in Figure 1 air quality data of 29 stations from İstanbul are selected for the study. In the figure, white circles symbolizes for IoT devices, red nodes symbolizes fog devices and the yellow node symbolizes the cloud center.

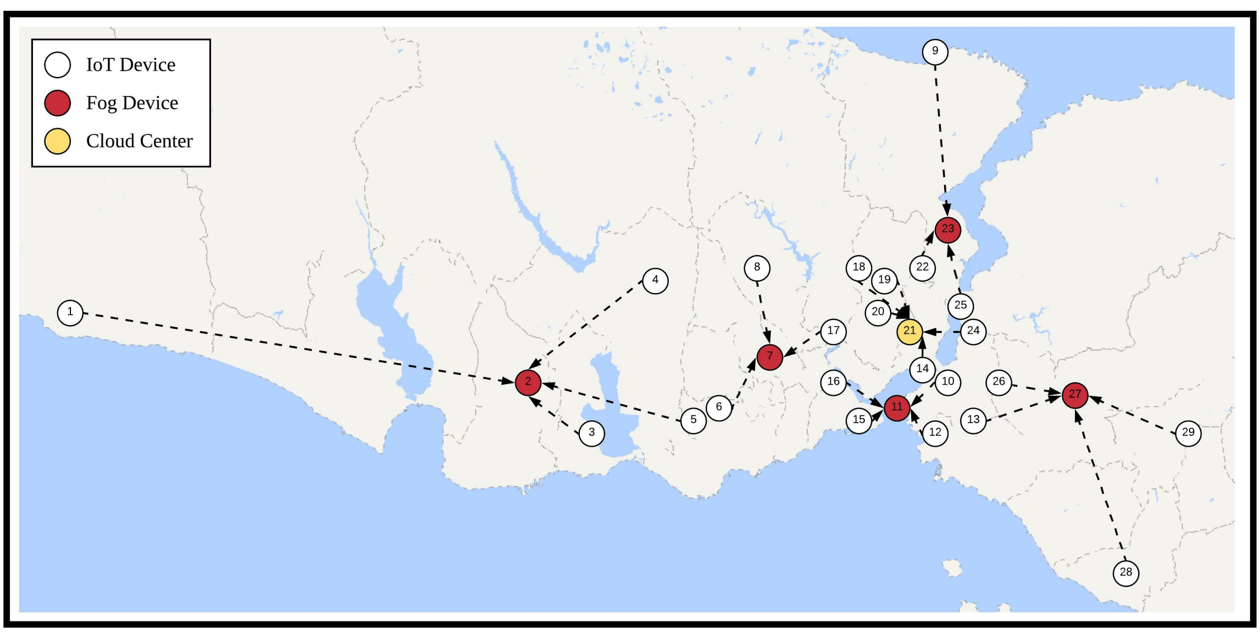

Figure 1. The network structure.

Pollutant concentrations of the year 2019 are downloaded from the portal of the ministry [45]. The last day of the year is discarded and total duration is decomposed into hourly intervals. In total, 8736 time-slices are acquired. Data densities of time slices' (rows) and pollutants (columns) are calculated. Pollutants and time slices with more than 10% missing data are discarded. Concentrations lower than zero or higher than a specified upper bound are labelled as "outlier". Outlier values are replaced with NaN, and all missing values are imputed with KNNimpute algorithm [46].

Two types of prediction mechanisms are implemented, namely cross-pollutant and spatial prediction. Cross-pollutant prediction (CPP) is performed at each station to acquire a local prediction. To predict a pollutant's concentration, timely features $(season, DoW, HoD)$ and concentrations of all pollutants from last two hours are used. CPP is performed for each pollutant at every time slice. Therefore, at every time slice each station makes a prediction of all monitored pollutants for the next hour. As an example, performed CPP's of station 4 for CO is given below. $CPP(A, B, C)$ stands for station A's CPP for pollutant B at the $C'th$ time slice. Likewise, $M(D, E, F)$ stands for station $D's$ monitored concentration of pollutant E at the $F'th$ time slice. $S(G, H)$, $DoW(G, H)$ and $HoD(G, H)$ stands respectively for season information, day-of-week information and hour-of-day information for station G at the $H'th$ time slice.

- $CPP(4, CO, t) = Predict(S(4, t), DoW(4, t), HoD(4, t), M(4, CO, t-1), M(4, NO2, t-1), M(4, O3, t-1), M(4, PM10, t-1), M(4, SO2, t-1), M(4, CO, t-2), M(4, NO2, t-2), M(4, O3, t-2), M(4, PM10, t-2), M(4, SO2, t-2))$

Second type of prediction is spatial prediction (SP). SP aims to predict a pollutant's concentration using monitored concentrations of same pollutant across other stations. In this case, we also add timely features to enhance prediction accuracy. SP task is assigned to fog nodes, and as in CPP, SP is also performed at every time-slice using data from last two hours. Each fog node is responsible for prediction of a single specified pollutant for all stations that monitor the specified pollutant.

4. Proposed models

In conventional IoT networks, sensors collect data from the environment and send their data to a single server without any regard of the content or the type of application that request the data. Hence, especially for large-scale IoT networks, the volume of data transmitted among IoT devices may cause undesirable conditions such as congestion, higher transmission delays, and reduced QoS levels. The concept of prioritization is shown to be a proper solution for dealing with heterogeneous and ubiquitous IoT data [47]. Therefore, this paper proposes two novel content-based data forwarding models for fog-based IoT networks using SDN. The proposed high-level system architecture is shown in Figure 2. The architecture employs SDN and FC for IoT networks and consists of components of them. The proposed models assume that the IoT network has several fog servers (nodes), and each one of them is dedicated to a certain data type and/or application. In addition, in the models, SDN controllers are responsible for arrival of packets to their respective fog servers. The proposed models are implemented and evaluated using the realistic network flow generated from air quality measurements of Istanbul province. To make it easier to understand the proposed models and evaluation scenario, detailed information about the data and generated network flow is given in the next section.

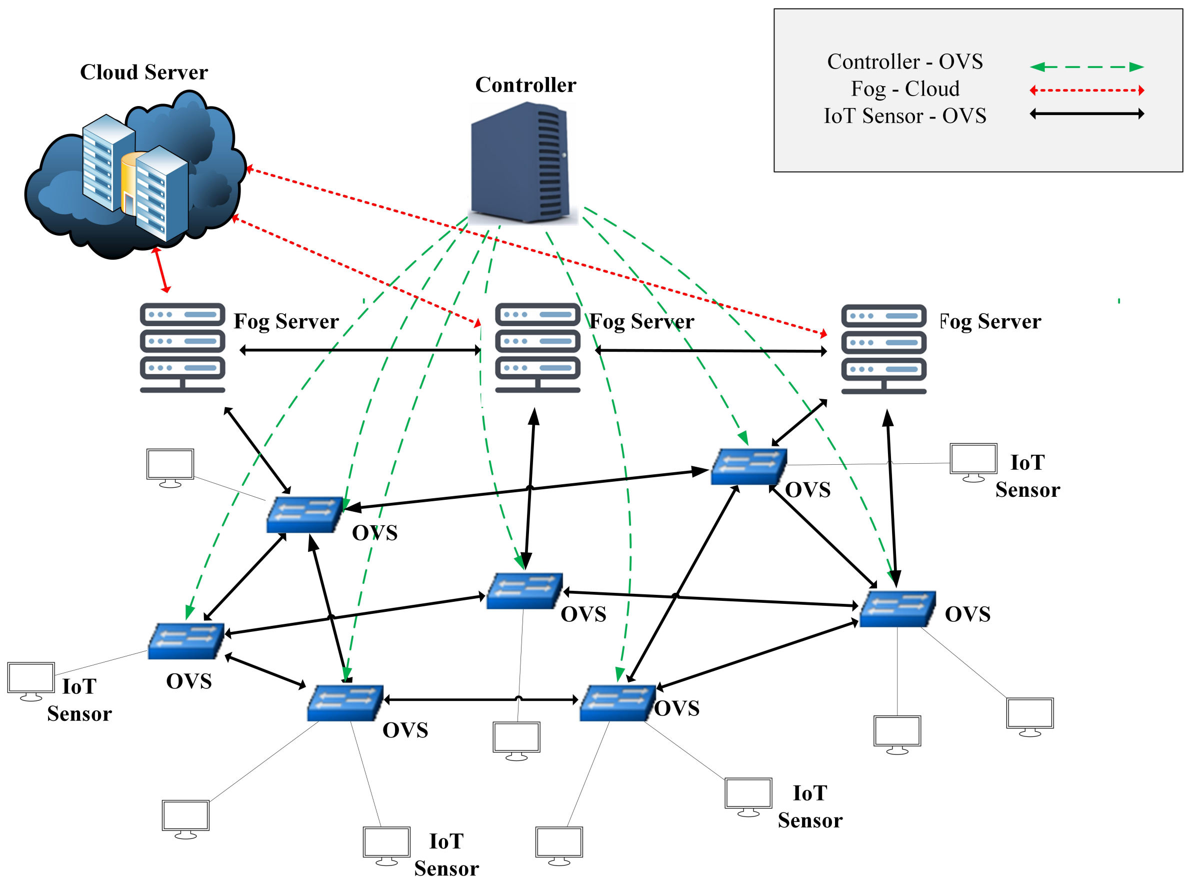

Figure 2. Proposed architecture of SDN based IoT/FC.

4.1. Data

The data is assumed to be collected according to the structure that is shown in Figure 1. The sensor readings are classified according to their impact values. Higher pollutant concentrations pose a higher risk for people. For this reason, an impact level table is generated using pollutant levels specified in [45]. As shown in Table 1, the value ranges of the data determining the air quality are divided into 5 levels. Level 1 represents the lowest effect whereas level 5 exemplifies the highest one. The levels 1 and 2 are considered as safe levels. Level three is moderate air quality and points out that people with respiratory and cardio-vascular health problems can be affected by the air pollution. Levels 4 and 5 are considered dangerous for all people and can cause long-lasting health problems[1] [45]. The data packet priority levels are assigned parallel to their impact on health.

The sensor readings in the created data set is classified according to Table 1, and the result of the classification process is given in Table 2. Table 2 reveals that the critical pollutants that exceed threshold values are PM10, ozone (O3) and NO2. Among these, PM10 seems to be the most critical pollutant. Because, as can be seen from Table 2, the majority of sensor readings that exceed determined threshold values are PM10 readings for all fog nodes. The pollutants CO and SO2 seem to have a very low number of readings that exceed their respective thresholds. The effects of NO2 and Ozone on the other hand, are not constant and change

[1]Air Quality Now: CITEAIR Project funded by The European Union [online]. Website https://www.airqualitynow.eu/ [Accessed: 25.09.2021]

Table 1. Priority levels of the data.

	Level 1		Level 2		Level 3		Level 4		Level 5	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
CO	0	4999	5000	9999	10000	14999	15000	19999	20000	∞
NO2	0	49	50	99	100	149	150	199	200	∞
O3	0	59	60	119	120	179	180	239	240	∞
PM10	0	24	25	49	50	74	75	99	100	∞
SO2	0	174	175	349	350	524	525	699	700	∞

Table 2. Numbers of packets in the data.

Node	**Pollutants**	**Priority level**				
		Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**
Fog Node 1	CO	165.878	0	0	5	0
	NO2	43.550	17.884	4.676	406	31
	O3	31.570	876	0	0	18
	PM10	55.918	19.289	4.677	1.340	841
	SO2	49.497	0	0	0	1
	Total (of Fog Node)	346.413	38.049	9.353	1.751	891
Fog Node 2	CO	32.546	0	0	1	0
	NO2	159.883	61.251	9.837	802	173
	O3	29.459	2.486	8	0	9
	PM10	36.275	9.541	1.904	482	220
	SO2	0	0	0	0	0
	Total (of Fog Node)	258.163	73.278	11.749	1.285	402
Fog Node 3	CO	16.680	0	0	0	0
	NO2	52.497	11.339	2.131	245	27
	O3	118.354	24.951	580	5	763
	PM10	37.968	14.235	7.351	3.088	2.757
	SO2	65.402	0	0	0	0
	Total (of Fog Node)	290.901	50.525	10.062	3.338	3.547
Fog Node 4	CO	49.393	0	0	1	0
	NO2	62.232	3.630	103	0	6
	O3	39.598	8.671	361	2	104
	PM10	124.205	105.986	37.252	12.644	11.197
	SO2	65.271	1	0	0	2
	Total (of Fog Node)	340.699	118.288	37.716	12.647	11.309
Fog Node 5	CO	24.646	0	0	0	0
	NO2	42.384	6.558	552	15	66
	O3	35.663	4.717	175	1	54
	PM10	56.049	16.605	5.436	1.947	1.720
	SO2	202.509	0	0	0	9
	Total (of Fog Node)	361.251	27.880	6.163	1.963	1.849
Fog Node 6	CO	66.631	0	0	1	0
	NO2	50.800	14.333	1.316	38	1
	O3	28.816	4.032	13	0	3
	PM10	56.502	13.535	6.876	2.575	2.725
	SO2	33.009	0	0	0	3
	Total (of Fog Node)	235.758	31.900	8.205	2.614	2.732
Total	Total (of Network)	1.833.185	339.920	83.248	23.598	20.730

between fog devices. At Fog Node 1, both NO2 and ozone have limited effect on air quality at moderate level, very limited effect at dangerous levels. At Fog Node 2, NO2 is a major pollutant and ozone has very little effect. At Fog Note 3, NO2 and ozone have a strong effect on air quality at safe to moderate levels. At the dangerous levels, only ozone has a strong impact on air quality. At Fog Node 4, both NO2 and ozone have very limited effect on air quality compared to the main pollutant PM10. Similarly, at the fog nodes 5 and 6, NO2 and ozone have very limited effect on air quality at moderate and dangerous levels. The analysis reveals that the main pollutant that causes a decrease in air quality in Istanbul is PM10 by far. PM10 is followed by NO2 and ozone, at specific regions of İstanbul and mostly in safe to moderate air quality levels. The pollutants SO2 and CO have a very limited effect on air quality in İstanbul. As mentioned, the proposed priority based model, namely PCBF-SDN, changes prioritization of packages based on how critical the package is. PCBF-SDN handles packages based on criticality without any regard of the pollutant type. Therefore, it would be safe to assume that the most of the packages with priority is going to be PM10 sensor readings, followed by packages that carry sensor readings of NO2 and Ozone. CO and SO2 packages will be very rare among the packages with high priority.

As seen from Table 2, in total 8734 hours of the network are simulated, and 2,300,681 packets are employed for the validation of the proposed models.

4.2. First model: content based forwarding via software defined network

In IoT systems, many sensors can be connected to a computer, and thousands of sensors can be connected to these systems. These systems are dynamic, and the needs may change rapidly over time. When requirements change, each computer, the sensors, and the switches should be optimized again. This creates a repeated and laborious workload in large systems. However, the routing is performed with the controllers' support in the switches in the proposed model. The entire system can be changed according to the needs with a simple central update in the controller software.

With CBF-SDN, the data obtained from the sensors are tagged according to the data type, then transmitted to the network. Tagging the data by type allows switches and controllers in the network to identify the data. The tags are known by the controller and are transmitted to SDN-enabled data plane devices. On the other hand, in the traditional network structure, the content of the data packet in the network layer is not known by the network devices, which communicate by looking at only certain bits. This is the most crucial difference that distinguishes the model we recommend from traditional ones.

The data obtained from the sensors are processed on the fog servers, and the result is transmitted to the main (cloud) server. In addition, distributed data processing can be performed by processing different sensors' data on each fog server. Consequently, the proposed CBF-SDN model provides advantages in network development by determining the location of fog servers, optimizing the network, selecting the shortest path, etc. The fact that the controller and switches recognize the content of the data in SDN-based networks delivers many advantages. These advantages can be listed as the following;

- In the emergency scenario, data that are lower than the specified threshold value are not transmitted on the network, and data types obtained from some sensors are not used in the tasks specified for fog servers. Thus, in the CBF-SDN model, the network performance is improved by preventing the transmission of the data that is not used or unimportant by the switch.

- As the labeling is done according to the data type, the data damaged by the sensors or environmental factors is prevented from being transmitted unnecessarily to the servers since the controller cannot identify it.

- Since content-based forwarding is performed, the controller performs parameter adjustments quickly and dynamically instead of the tremendous work required by accessing the sensors. These adjustments could be like changing the desired data type, transmitting specific data types to the fog servers, or making changes in the network infrastructure.

- Last but not least, sensors generate data at specified times. However, data produced at different times can be used in the same data processing. For instance, the CO sensor creates data every 5 min, but only data generated every 30 min is used for data processing. As the controller does not transmit unused data, the network performance is improved. Nevertheless, in CBF-SDN, the possibility that unsent data is essential for data processing has been ignored. Hence, in the second model that we proposed, a copy of only the important data (prioritized) is transmitted to the cloud server so that erroneous analysis results that due to data loss are reduced and avoided.

4.3. Second model: priority enabled content based forwarding via software defined network

The sensors measuring the air quality of the environment generate continuous data flow. Moreover, since some data types are observed to be more vital than others, as seen in Table 2, the values of the data are also significant in defining its priority as far as they impact the analysis results. Consequently, transferring data with a high influence on the results to a separate queue and granting privileges to this queue affects the analysis results and network performance. As a result, the CBF-SDN structure is extended with priority queues, and this novel architecture is called priority-enabled content-based forwarding via SDN (PCBF-SDN). To elaborate, the data types, which are higher than the specified threshold value or just significant, directly impact the analysis results. So, in PCBF-SDN, these significant data types are placed in a separate queue, and they are transmitted with higher priority and more bandwidth in the network. PBCF-SDN is an extension to CBF-SDN. Because of this reason, the flowchart of CBF-SDN and PBCF-SDN is given jointly in the Figure 3. Blue arrows in the figure are the ones shared in both models. Red ones are CBF-SDN only connections and similarly orange connections represent PBCF-SDN only connections. As shown in Figure 3, the controller initiates the communication between devices in the network. Furthermore, it has data type information of packets transmitted to the network. If the information of the transmitted packets is available in the flow tables of the switches, it is transmitted. Otherwise, the data type information is asked to the controller. Packages are modified according to the fog server that needs to be routed. Afterward, it is checked whether the data is greater than the specified threshold value, and whether it will be included in the priority queue is decided. Then, the data is transmitted to the server that needs to be processed. Air quality is decided by processing different data forwarded to the same fog servers. Finally, a copy of the priority queued data is sent directly to the main server. The purpose here is to make quick decisions on the main server in case of an emergency. PCBF-SDN makes faster decisions for potential emergency transmission, bypasses the fog server, and forwards the data directly to the main server. For example, when the CO rate is higher than 15,000, the life quality of the environment decreases considerably. Therefore, this data is transmitted to the fog server and the main server simultaneously. That ensures that decisions are taken more quickly in such emergencies.

As stated previously, PCBF-SDN is the extended version of the CBF-SDN model. For this reason, the PCBF-SDN model is explained with technical details in the rest of this study. As seen in the flow diagram, the proposed model is managed by the SDN controller. The communication of the devices in the network, the

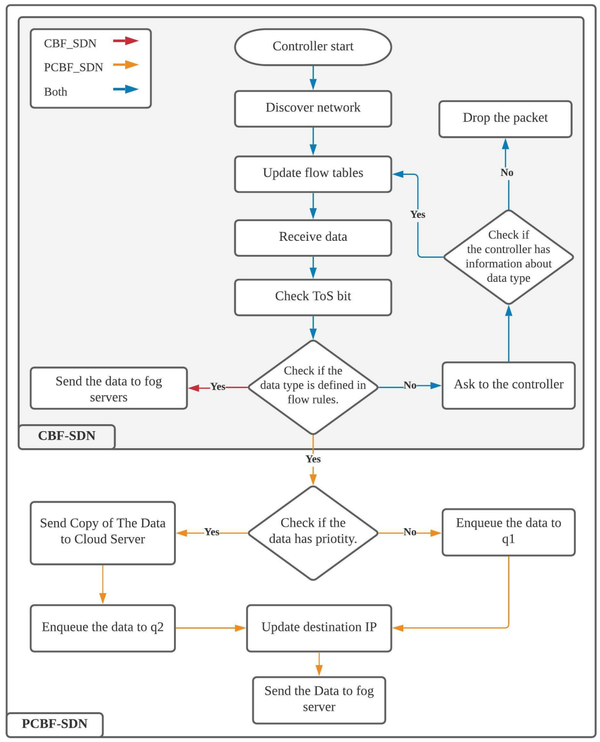

Figure 3. The flowchart of the proposed models.

transmission of the flow protocols to the network devices, and the forwarding of the packets are all carried out by this controller. Also, the model decides to which fog server the packets (based on the contents) are forwarded and whether a copy must be forwarded to the main server or not.

Yet, initially, when the controller discovers the network, only the connected devices can communicate. Therefore, there is no previous information about the packets in the ow tables on the switches before transmitting the data. So, the data obtained from the sensors are labeled without being transmitted to network devices. This labeling can be happening by changing the type of service (ToS) bit values. As shown in Table 3, different ToS bit values are used according to each data type, and the controller formerly knew these values. The data is wrapped and routed to the switches, and as shown in Algorithm 1, the switches modify the destination IP according to the ToS bit and send this altered data to one of the queues. If the switch does not have any routing information in the flow table, it asks the controller.

Table 3. ToS bits of data types.

Data type	ToS bits
CO	0xa0
NO2	0xb0
O3	0xc0
PM10	0xd0
SO2	0xe0

The controller continuously transmits the necessary information to all switches and updates the flow tables. That ensures that the types of data contained in the packets are known by the other switches, which are able to forward the packets based on this information. After the data type is determined, the packet is sent to one of the two separate queues according to its priority. Queue-1 (q1) is the queue with prioritized data and has wider bandwidth than Queue-2 (q2) which has non-prioritized data. Since the proposed model is managed with SDN, routing can be easily updated according to the requests of data analysts. In other words, when the importance of some data changes according to the results of the data analysis, its priority is updated and reflected in the forwarding decisions with the controller's support.

Algorithm 1: OpenFlow rules of the proposed PCBF-SDN model.

```
Input: Data packet p, ToS bit of packet T, packet time out t, Flow table of switch ft
Output: Destination IP dest-ip, Queues q1 q2
initialization:
while t ≠ 0 do
    if ft has information about T then
        dest-ip = modify destination ip from ft
    end if
    ask Controller
    if Controller has information about packet then
        update all flow tables
    end if
    drop packet
end while
if p has priority then
    get data to q1
end if
get data to q2
Send p with new dest-ip
```

5. Experimental analysis

This section examines the effects of processing data obtained from IoT sensors on fog servers in SDN-enabled networks. The data gathered on six different fog servers are analyzed, and the results are forwarded to the main server. Data analysis operations are performed with the data collected on the main server, and the results are compared to those obtained by collecting and processing the data obtained from the sensors on a single server. As stated in the previous section, the data coming from the sensors are prioritized according to the effect on the analysis result. The implications of using the proposed queue structure and the impact of data prioritization on both network performance and data analysis are examined.

5.1. Experimental setup

In this section, we explain the experimental design followed in this study. We have three different models to compare one of them is traditional and the other two are proposed by this study. The traditional network is a network model that does not support by SDN. Switches are independently configured on the GNS3 platform. The priority of the data has not been examined. Data is not prioritized and transmitted in a single queue. The data obtained from the sensors were sent to the designated servers. The scenario in Figure 4 was applied precisely.

In the traditional IoT model, data from the sensors are transmitted directly to the cloud server. However, in the CBF-SDN model, the data from the sensors are routed according to their content and transmitted to the main server after their processing on the fog servers. Moreover, in the PCBF-SDN model, sensors' data are prioritized, then processed on fog servers by content-based forwarding, and finally transmitted to the main server. However, in the PCBF-SDN model, a copy of sensor data with high priority is directly transmitted to the main server without being processed.

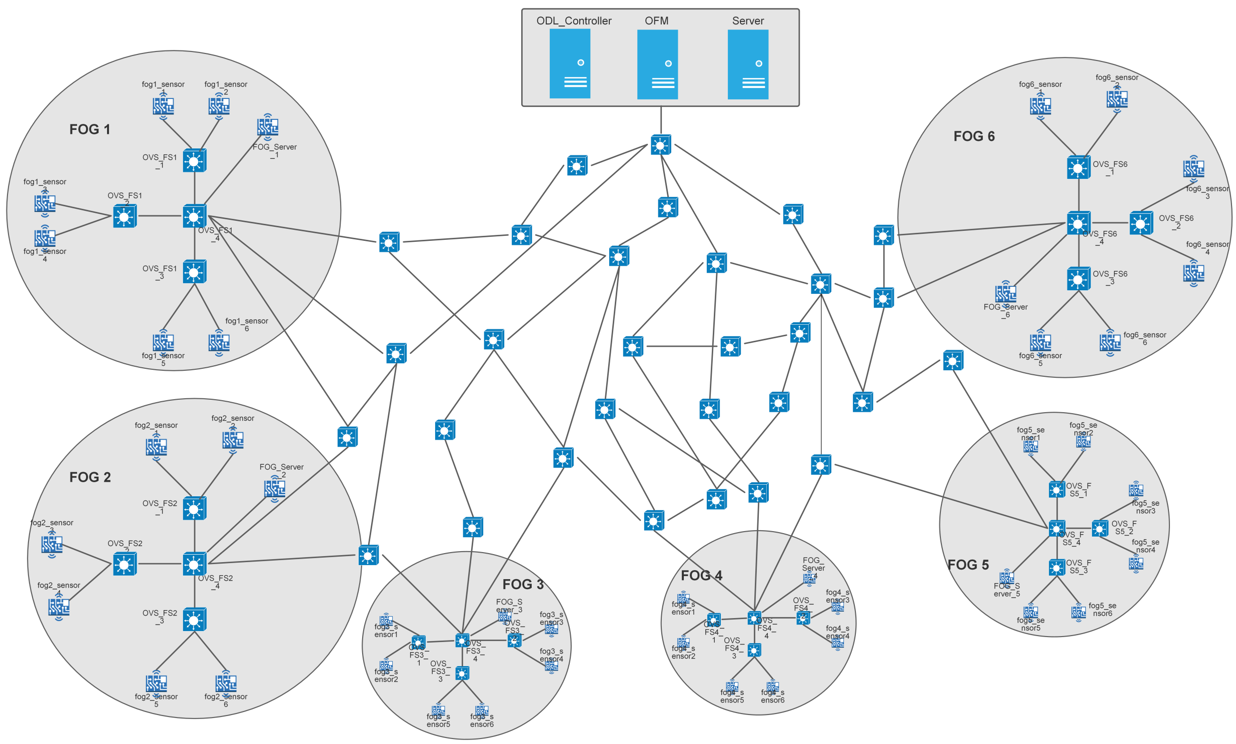

Figure 4. The generated network topology.

The test environment consists of 6 fog servers, 6 computers connected to each server, and several sensors connected to each computer. The simulation contains 70 Open vSwitch (OVS)[2] instance, a controller server, an OpenFlow Manager (OFM)[3] server, and the main server. The proposed models are simulated in the Graphical Network Simulator-3 (GNS3)[4] environment version 2.2.5. The sensors are installed in Docker containers[5] and have a Linux operating system. In addition, data is transmitted to the network by changing ToS bits with Netcat[6] utility software. The network topology is shown in Figure 4.

According to the literature [48–50], many SDN controllers are used, such as OpenDaylight (ODL), POX, Ryu, etc. In this study, we select the ODL controller to work with because it is highly compatible with OFM, OVS, and GNS platforms. ODL is installed on a virtual machine that has a Windows operating system. This virtual machine is integrated with GNS3 and passes the management of the network to the ODL controller.

[2] Open vSwitch [online]. Website https://www.openvswitch.org/ [accessed: 15.11.2021]
[3] Openflow Manager [online]. Website https://github.com/CiscoDevNet/OpenDaylight-Openflow-App [accessed: 15.11.2021]
[4] GNS3 Software [online]. Website https://docs.gns3.com/ [accessed: 15.11.2021]
[5] Docker [online]. Website https://docs.docker.com/ [accessed: 15.11.2021]
[6] Netcat [online]. Website https://nmap.org/ncat/ [accessed: 15.11.2021]

The ODL controller identifies the devices and communication paths in the network and enables the devices in the network to communicate.

The ODL controller decides the next hop for each type of data on each switch. Next, it sends flow rules to all switches to enforce the forwarding decisions. After that, each switch performs forwarding according to the rules on its flow tables. Then, when packets arrive at a switch, they are matched to one of the rules in the flow tables based on the predefined ToS bit patterns that represent the data types. The corresponding forwarding action defined in the rule involves queuing the packet on the correct queue based on the priority of its data type. If the packet cannot be matched to an existing rule, the packet is sent to the controller. At this point, the controller finds out the data type, installs a new rule on the switches, and accordingly modifies the ToS bits and destination IP.

The specifications of the test environment in this study can be summarized as follows. We employed a server that has a 64-bit Linux operating system, 3.4 GHz processor, and 32 GB RAM. In the simulated topology designed with GNS3, there are 70 OVS switches and 36 virtual computers running on the Docker container, and 3 virtual servers on VMware. Each sensors produces 90 packets of data per hour, with an average packet size of 75 bytes. While bandwidth is determined as 10 Mbit/s in the CBF-SDN model, 6 Mbit/s for q1 and 4 Mbit/s for q2 are defined for the PCBF-SDN model.

5.2. Evaluation metrics

This subsection describes the metrics utilized to assess the three models discussed in this study. We first discuss the threshold values selected for the priority queue for every single data type. Table 4 shows the threshold values defined for the priority queued data in the data set described earlier. The threshold values are determined in parallel to the impact levels given in Table 1 which uses critical pollutant concentrations defined in Common Air Quality Index (CAQI) of European Union[7] and defined by Turkish Ministry of Environment, Urbanization and Climate Change [45].

According to the impact values specified in Table 1 and the threshold values defined in Table 4, the number of priority data is 245,451 and the total number of data is 2,300,681. As a result of the statistical analysis of the complete data set, approximately 10,67% of the sensors data is determined as priority data.

Table 4. Threshold values of important data.

Data type	Threshold values
CO	>15000
NO2	>100
O3	>120
PM10	>25
SO2	>175

Traditional IoT networks and the proposed models are evaluated based on several prevalent metrics in the literature for such scenarios. These metrics are:

- Throughput, which is the total packet size transmitted and received per unit time,
- Average delay, which is the time it takes from the source to the destination,

[7]Air Quality Now: CITEAIR Project funded by The European Union [online]. Website https://www.airqualitynow.eu/ [Accessed: 25.09.2021]

- Average loss rate is the ratio of the number of packets that cannot reach the target to the total number of packets.

The results obtained from the simulation experiments regarding these three metrics are fully presented and discussed in the next section.

5.3. Results and discussion

The data obtained to measure air quality is simulated like an IoT network. As a result of the analysis made, the priority values of the data are determined. The proposed model is compared to traditional networks in terms of the average delay and throughput with this data.

As shown in Table 5, an average delay of 1.1 ms is observed in traditional networks. In the CBF-SDN model, the average delay of 1.02 ms is observed. Moreover, in the PCBF-SDN model, the average delay of priority data is observed as 0.8 ms, while nonpriority data is 1.3 ms, and the average delay of whole data is 1.23 ms. The benefits of having less latency on priority data are explained in previous sections. These delays are most affected by the bandwidth allocated to the priority queue and the ratio of priority data to total data.

Table 5. Average delay of proposed models.

Model		**Average delay (ms)**
Traditional IoT		1,10
CBF-SDN		1,02
PCBF-SDN	Priority data	0,80
	Nonpriority data	1,30
	All data	1,23

The rate of missing data that the controller could not identify is approximately 3% in the CBF-SDN model. Since the data from the sensors are analyzed on the fog server, that affects the number of packets transmitted to the network. Therefore, the throughput values in the CBF-SDN model are lower than the traditional IoT's. The main reason why our proposed model (CBF-SDN) has less average delay is our proposed model's ability to identify the corrupted/missing data and eliminate them. As the data transmitted to the network is decreasing, the throughput is decreasing, and therefore the average latency is reducing.

As shown in Table 6, an average throughput of 2.03 Mbits and the number of packets per second 15 is observed in traditional networks. However, in the CBF-SDN model, the average throughput of 1.86 Mbits and the number of packets per second 13 is observed. Moreover, in the PCBF-SDN model, the average throughput of 1.69 Mbits and the number of packets per second 17 is seen. As shown in Table 6, the throughput values and number of packets of the three models are compared. As can be noticed, throughput values in the CBF-SDN model are lower than those in the traditional IoT. The sensors' data are analyzed on the fog server, and this changes the number of packets transmitted to the network. In other words, in the proposed models, the data is first processed on fog servers. Then, the corrupted/missing data from the sensors are eliminated, so the average number of packets sent to the network is decreased. Moreover, in the PCBF-SDN model, the total bandwidth is divided into priority and nonpriority queues, and 40% is defined for the priority queue. So the throughput is decreased proportionally with the rate of bandwidth. The total number of packets increases because they are transmitted a copy of data to the main server, as explained in the previous section (just for prioritized data).

Table 6. Average throughput and number of packets of proposed models.

Models	Throughput(Mbits)	Number of packets
Traditional IoT	2.03	15
CBF-SDN	1.86	13
PCBF-SDN	1.69	17

As shown in Figure 5, the average delay of proposed models is compared according to the bandwidth allocated to the priority queue. The total bandwidth is divided into priority and nonpriority queues. In Figure 5, the bandwidth allocated to the priority queue is given as a percentage. Therefore, the remaining part is allocated to the nonpriority queue. Previously, it was stated that the ratio of priority data to total data was 10.67%. As the bandwidth allocated to the priority queue increases, the latency of the priority data decreases while the average latency increases. It has been observed that there is no improvement in latency of priority data in case the ratio of bandwidth allocation to the priority queue surpasses 40%, which indicates the ideal ratio for the proposed model. In order to employ the proposed model most appropriately, the optimum bandwidth should be determined for the priority queue.

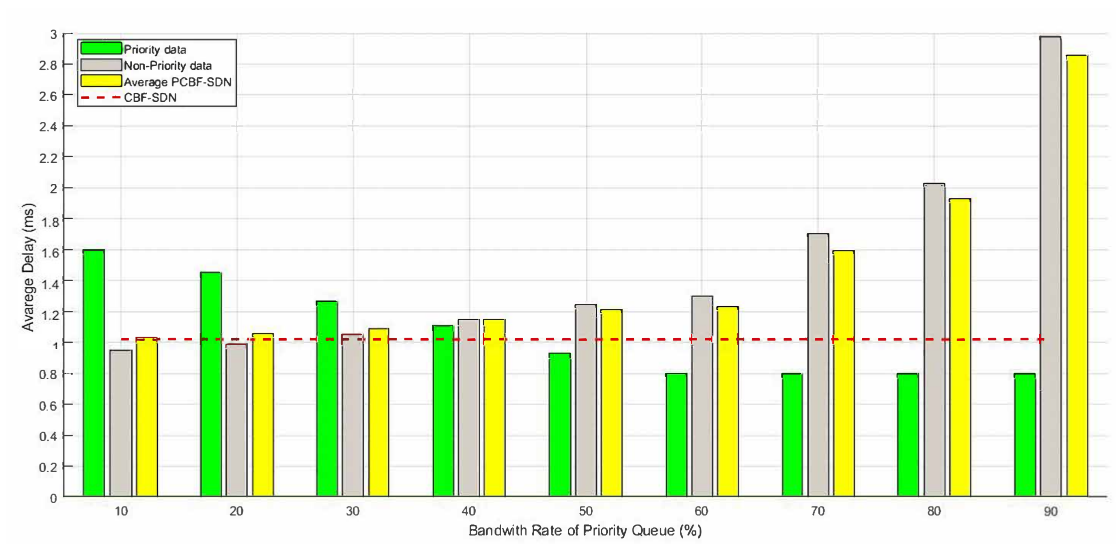

Figure 5. Relationship between average delay and bandwidth rate of priority queue.

With PCBF-SDN, the average delay in the transmission of critical data is reduced. In other words, the critical data that is revealed by data analysis can be transmitted faster. Accordingly, this positively affects the results in the realtime data processing. However, the rapidly increasing data load negatively affects the performance of the network. For this reason, an SDN controller is employed in our proposed model. By utilizing OpenFlow rules, network performance is increased by ensuring that only essential data is injected to the network. On the other hand, the data that is not used in data analysis is discarded. The rules can be remotely, dynamically, and easily updated to the system according to requirements.

6. Conclusion

In this study, a FC-based architecture is proposed to transmit sensor data in SDN-enabled IoT networks efficiently. Two data forwarding models are proposed in this architecture. The first model (CBF-SDN) makes use of SDN to perform content-based data forwarding. The second model (PCBF-SDN) focuses on delivery of critical data with low transmission delay. To this end, PCBF-SDN extends CBF-SDN by employing the priority concept during data forwarding. The performance evaluation results show that the proposed FC-based architecture model reduces delay and bandwidth consumption in large-scale IoT networks. Moreover, utilizing the priority concept in SDN-based data forwarding ensures timely delivery of critical data.As our future plans, we intend to improve the performance further by dynamically assigning bandwidth to the queues and increasing the network's performance by producing solutions to the controller location problem. Also, we are going to focus on improving our models with the aim of reducing the overhead of controller-switch communication.

Acknowledgment

This work is partially supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under the grant number 118E212.

References

[1] Mahmud R, Kotagiri R, Buyya R. Fog Computing: A Taxonomy, Survey and Future Directions. In: Di Martino B, Li KC, Yang L, Esposito A (editors). Internet of Everything. Internet of Things (Technology, Communications and Computing). Springer, Singapore, 2018, pp. 103-130. doi:10.1007/978-981-10-5861-5_5

[2] Yousefpour A, Ishigaki G, Jason JP. Fog computing: Towards minimizing delay in the internet of things. In: 2017 IEEE international conference on edge computing (EDGE); Honolulu, HI, USA; 2017. pp.17-24. doi:10.1109/IEEE.EDGE.2017.12

[3] Samann FEF, Zeebaree SR, Askar S. IoT provisioning QoS based on cloud and fog computing. Journal of Applied Science and Technology Trends 2021; 2 (01): 29-40. doi:10.38094/jastt20190

[4] Hu P, Dhelim S, Ning H, Qiu T. Survey on fog computing: architecture, key technologies, applications and open issues. Journal of network and computer applications 2017; 98: 27-42. doi:10.1016/j.jnca.2017.09.002

[5] García-Pérez CA, Merino P. Experimental evaluation of fog computing techniques to reduce latency in LTE networks. Transactions on Emerging Telecommunications Technologies 2018; 29 (4): 1-17. doi:10.1002/ett.3201

[6] Siasi N, Jaesim A. Priority-aware SFC provisioning in fog computing. In: IEEE 2020 17th Annual Consumer Communications & Networking Conference (CCNC); Las Vegas, USA; 2020. pp. 1-6. doi:10.1109/CCNC46108.2020.9045275

[7] Okay FY, Ozdemir S. Routing in fog-enabled IoT platforms: A survey and an SDN-based solution. IEEE Internet of Things Journal 2018; 5(6): 4871-4889. doi:10.1109/JIOT.2018.2882781

[8] Eki RL, Randy L, Kumar S. Method and system for priority based routing. U.S. Patent No. 8,014,404. 6 Sep. 2011.

[9] Kreutz D, Ramos FM, Verissimo PE, Rothenberg CE, Azodolmolky S et al. Software-defined networking: A comprehensive survey. Proceedings of the IEEE 2014; 103 (1): 14-76. doi:10.1109/JPROC.2014.2371999

[10] Kirkpatrick K. Software-defined networking. Communications of the ACM 2013; 56 (9): 16-19. doi:10.1145/2500468.2500473

[11] Alomari A, Subramaniam SK, Samian N, Latip R, Zukarnain Z. Resource Management in SDN-Based Cloud and SDN-Based Fog Computing: Taxonomy Study. Symmetry 2021; 13 (5): 734-764. doi:10.3390/sym13050734

[12] Lara A, Quesada L. Priority-based routing in a campus network using SDN. IN: IEEE 2017 9th Latin-American Conference on Communications (LATINCOM); Guatemala City, Guatemala; 2017. pp. 1-6. doi:10.1109/LATINCOM.2017.8240172

[13] Oh B, Vural S, Wang N, Tafazolli R. Priority-Based Flow Control for Dynamic and Reliable Flow Management in SDN. IEEE Transactions on Network and Service Management 2018; 15 (4): 1720-1732. doi:10.1109/TNSM.2018.2880517

[14] Lu Y, Fu B, Xi X, Zhang Z, Wu H. An SDN-based flow control mechanism for guaranteeing QoS and maximizing throughput. Wireless Personal Communications 2017; 97 (1): 417-442. doi:10.1007/s11277-017-4512-9

[15] Shinkuma R, Yamada Y, Sato T, Oki E. Flow control in SDN-Edge-Cloud cooperation system with machine learning. In: IEEE 2020 40th International Conference on Distributed Computing Systems (ICDCS); Singapore, Singapore; 2020. pp. 1304-1309. doi:10.1109/ICDCS47774.2020.00169

[16] Deng GC, Wang K. An application-aware QoS routing algorithm for SDN-based IoT networking. In: IEEE 2018 Symposium on Computers and Communications (ISCC); Natal, Brazil; 2018. pp. 186-191. doi:10.1109/ISCC.2018.8538551

[17] Diro AA., Reda HT, Chilamkurti N. Differential flow space allocation scheme in SDN based fog computing for IoT applications. Journal of Ambient Intelligence and Humanized Computing 2018: 1-11. doi:10.1007/s12652-017-0677-z

[18] Bardalai P, Medhi N, Bargayary B, Saikia DK. OpenHealthQ: OpenFlow based QoS management of Healthcare Data in a Software-Defined Fog environment. In: IEEE 2021 International Conference on Communications(ICC); 2021. pp. 1-6. https://doi.org/10.1109/ICC42927.2021.9500637

[19] Ghazi MU, Khattak MAK, Shabir B, Malik AW, Ramzan MS. Emergency message dissemination in vehicular networks: A review. IEEE Access 2020; 8: 38606-38621. doi:10.1109/ACCESS.2020.2975110

[20] Kadhim AJ, Seno SAH. Energy-efficient multicast routing protocol based on SDN and fog computing for vehicular networks. Ad Hoc Networks 2019; 84: 68-81. doi:10.1016/j.adhoc.2018.09.018

[21] Secinti G, Darian PB, Canberk B, Chowdhury KR. SDNs in the sky: Robust end-to-end connectivity for aerial vehicular networks. IEEE Communications Magazine 2018; 56 (1): 16-21. doi:10.1109/MCOM.2017

[22] Atlam HF, Walters RJ, Wills GB. Fog computing and the internet of things: a review. big data and cognitive computing. Big data and cognitive computing 2018; 2(2): 10-28. doi:10.3390/bdcc2020010

[23] Beck MT, Werner M, Feld S, Schimper S. Mobile edge computing: A taxonomy. In: Sixth International Conference on Advances in Future Internet (AFIN); Lisbon, Portugal; 2014. pp. 48-55. doi:10.1.1.670.9418

[24] Zhang C. Design and application of fog computing and Internet of Things service platform for smart city. Future Generation Computer Systems 2020; 112: 630-640. doi:10.1016/j.future.2020.06.016

[25] Okay FY, Ozdemir S. A fog computing based smart grid model. In: IEEE 2016 international symposium on networks, computers and communications (ISNCC); Yasmine Hammamet, Tunisia; 2016. p. 1-6. doi:10.1109/ISNCC.2016.7746062

[26] Qin H, Zhang H. Intelligent traffic light under fog computing platform in data control of real-time traffic flow. The Journal of Supercomputing 2021; 77(5): 4461-4483. doi:10.1007/s11227-020-03443-3

[27] Ning Z, Huang J, Wang X. Vehicular fog computing: Enabling real-time traffic management for smart cities. IEEE Wireless Communications 2019; 26(1): 87-93. doi:10.1109/MWC.2019.1700441

[28] Negash B, Gia TN, Anzanpour A, Azimi I, Jiang M et al. Leveraging fog computing for healthcare iot. In: Rahmani A, Liljeberg P, Preden JS, Jantsch A (editors). Fog Computing in the Internet of Things. Springer, Cham, 2018. doi:10.1007/978-3-319-57639-8_8

[29] Pop P, Zarrin B, Barzegaran M, Schulte S, Punnekkat S et al. The FORA fog computing platform for industrial IoT. Information Systems 2021; 98: 101727. doi:10.1016/j.is.2021.101727

[30] Alzoubi YI, Osmanaj VH, Jaradat A, Al-Ahmad A. Fog computing security and privacy for the Internet of Thing applications: State-of-the-art. Security and Privacy 2021; 4 (2): e145. doi:10.1002/spy2.145

[31] Mekki T, Jabri I, Rachedi A, Chaari L. Software-defined networking in vehicular networks: A survey. Transactions on Emerging Telecommunications Technologies 2021: e4265. doi:10.1002/ett.4265

[32] Tomovic S, Yoshigoe K, Maljevic I, Radusinovic I. Software-defined fog network architecture for IoT. Wireless Personal Communications 2017; 92 (1): 181-196. doi:10.1007/s11277-016-3845-0

[33] Nunes BAA, Mendonca M, Nguyen XN, Obraczka K, Turletti T. A survey of software-defined networking: Past, present, and future of programmable networks. IEEE Communications Surveys & Tutorials 2014; 16 (3): 1617-1634. doi:10.1109/SURV.2014.012214.00180

[34] McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L et al. OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review 2008; 38 (2): 69-74. doi:0.1145/1355734.1355746

[35] Rawat DB, Reddy SR. Software defined networking architecture, security and energy efficiency: A survey. IEEE Communications Surveys & Tutorials 2016; 19 (1): 325-346. doi:0.1109/COMST.2016.2618874

[36] Benzekki K, El Fergougui A, Elbelrhiti EA. Software-defined networking (SDN): a survey. Security and communication networks 2016; 9 (18): 5803-5833. doi:0.1002/sec.1737

[37] Lv Z, Xiu W. Interaction of edge-cloud computing based on SDN and NFV for next generation IoT. IEEE Internet of Things Journal 2019; 7 (7): 5706-5712. doi:1109/JIOT.2019.2942719

[38] Cicioğlu M, Çalhan A. Energy-efficient and SDN-enabled routing algorithm for wireless body area networks. Computer Communications 2020; 160: 228-239. doi:0.1016/j.comcom.2020.06.003

[39] Al-Hubaishi M, Çeken C, Al-Shaikhli A. A novel energy-aware routing mechanism for SDN-enabled WSAN. International Journal of Communication Systems 2019; 32 (17): e3724. doi:1002/dac.3724

[40] Baktir AC, Ozgovde A, Ersoy C. How can edge computing benefit from software-defined networking: A survey, use cases, and future directions. IEEE Communications Surveys & Tutorials 2017; 19 (4): 2359-2391. doi:0.1109/COMST.2017.2717482

[41] Huang L, Li G, Wu J, Li L, Li J et al. Software-defined QoS provisioning for fog computing advanced wireless sensor networks. In: IEEE 2016 Sensors; Orlando, FL, USA; 2016. p. 1-3. doi:10.1109/ICSENS.2016.7808814

[42] Kadhim AJ, Naser JI. Proactive load balancing mechanism for fog computing supported by parked vehicles in IoV-SDN. China Communications 2021; 18 (2): 271-289. doi:23919/JCC.2021.02.019

[43] Alizadeh MR, Khajehvand V, Rahmani AM, Akbari E. Task scheduling approaches in fog computing: A systematic review. International Journal of Communication Systems 2020; 33 (16): e4583.doi:0.1109/TC.2016.2536019

[44] Muthanna A, Ateya AA, Khakimov A, Gudkova I, Abuarqoub A et al. Secure and reliable IoT networks using fog computing with software-defined networking and blockchain. Journal of Sensor and Actuator Networks 2019; 8 (1): 15. doi:03390/jsan8010015

[45] SIM (Sürekli Izleme Merkezi / Continuous Monitoring Center): Minister of Environment, Urbanisation and Climate Change. https://sim.csb.gov.tr/Services/AirQuality (Accessed on: 25.09.2021)

[46] Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T et al. Missing value estimation methods for DNA microarrays. Bioinformatics 2001; 17 (6): 520–525. doi:0.1093/bioinformatics/17.6.520

[47] Inagaki Y, Shinkuma R, Sato T, Oki E. Prioritization of Mobile IoT Data Transmission Based on Data Importance Extracted From Machine Learning Model. IEEE Access 2019; 7: 93611-93620. doi:0.1109/ACCESS.2019.2928216

[48] Stancu AL, Halunga S, Vulpe A, Suciu G, Fratu O et al. A comparison between several Software Defined Networking controllers. In: IEEE 2015 12th international conference on telecommunication in modern satellite, cable and broadcasting services (TELSIKS); Nis, Serbia; 2015. pp. 223-226.doi:0.1109/TELSKS.2015.7357774

[49] Salman O, Elhajj IH, Kayssi A, Chehab A. SDN controllers: A comparative study. In: IEEE 2016 18th Mediterranean Electrotechnical Conference (MELECON); Limassol, Cyprus; 2016. pp. 1-6. doi:0.1109/MELCON.2016.7495430

[50] Khondoker R, Zaalouk A, Marx R, Bayarou K. Feature-based comparison and selection of Software Defined Networking(SDN) controllers. In: IEEE 2014 World congress on computer applications and information systems (WCCAIS); Hammamet, Tunisia; 2014. pp. 1-7. doi:0.1109/WCCAIS.2014.6916572