International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Congestion Avoiding Approach Using Optimization Algorithm for IoT Web Services

Saima Aleem[1*], Shish Ahmad[2]

Submitted:10/03/2024 **Revised**: 25/04/2024 **Accepted**: 02/05/2024

Abstract: The Internet of Things, or IoT, is a technology that uses real-time data transfer via sensors to monitor remote sites and other systems. In this study, the "Thingspeak" web service, which is based on the Internet of Things, is examined as an open access API service that serves as a host for a range of sensors to monitor data received from sensors at the cloud level and transfer the data to the MATLAB platform at the designated channel ID with an API key. It is suggested to use an alternative optimization strategy for handling congestion and load balancing with Thingspeak-based IoT web services. This would improve network performance by distributing traffic load along the best paths possible to prevent traffic congestion.

Keywords: *Thingspeak, IoT, cloud services, Quality of Service (QoS), Optimization.*

1. Introduction

Demand for internet has increased for data exchange based services. Internet of things (IoT) has proved as an efficient application for connecting physical devices via internet & sensors. For efficient IoT services the proper monitoring and recording of signal is desired significantly. Sensing unit recording variation in humidity, temperature pressure etc to monitor the respective quantity in terms of fabricated values of current and voltage parameters. The IoT has brought a rapid change in modern world and becoming integral part of daily tasks [1]. The big organization has associated the IoT application in the most of the operations due to its significance in different areas [2]. Two decades before radio frequency identification (RFID) system used for performing routing task in surveillance, security, supervision, transport and healthcare based activities. Presently location & time of any the object may be tracked simultaneously by IoT services. Such application is creating multiple research ideas and scopes that are ensuring that by 2030, the computers may be capable of giving performance equivalent to humans in different types tasks management operations.

Currently, there are more Internet-connected gadgets than people on the planet [3]. These gadgets help users make wise judgments and provide helpful information about their surroundings. Machine-to-machine (M2M) communication has made it possible for intelligent end-user applications that rely on device-to-device communication without human interaction [4]. IoT integration has enabled more efficient and beneficial network connection setup by bringing items, people, and processes together [5]. The Internet of Things (IoT) is combining many technologies to create cutting-edge applications for linked individuals, computer systems, and common things. Network kinds and communication standards viewpoints can be used to achieve the implications related to IoT ideas. Low energy Bluetooth, RFID, ZigBee, and IPv6 are among the communication protocols viewpoints that are included in low-power personal area networks (LoWPANs) and wireless-fidelity (Wi-Fi), which uses the 5.850–5.925 Giga Hertz band for wireless access in systems related to vehicle communication. Wireless sensor networks (WSNs), cellular networks, and low-power wide-area networks serve as the foundation for the various network types [6, 7]. The wide number of wireless devices in the M2M network that can communicate independently at low data rates and low power modes sets it apart from conventional methods. M2M nodes often run on batteries and produce a lot of data that needs to be analyzed, saved, and presented in an understandable way [8].

Numerous applications (healthcare monitoring, surveillance, natural disaster, transportation, and smart cities) make advantage of the deployment of sensor nodes. But until the related data is shared over the Internet, its full potential remains unrealized [9].

2. Related Works

Prior to introduction of the proposed work of this article, the related research is explored for providing a review on the interoperability of programmable network concepts on cloud-based LoWPANs. In this section the networking schemes are reviewed for describing the impacts on performance of different types of network applications.

[1] *Integral University ,Lucknow, INDIA*
ORCID ID : 0000-0003-2030-7344
[2] *Integral University ,Lucknow, INDIA*
ORCID ID : 0000-0002-3347-7724
** Corresponding Author Email: saimaaleem@kmclu.ac.in*

Mahmud *et al.* [10] recognized the efforts that leverage association of concept of SDN for WSNs. They proposed a method known as Flow-Sensor, focused for tackling the problems associated to WSN and leading to a considerable cloud-based network performance enhancement by concept of network virtualization. The reachability of Flow-Sensor is observed to be more than the typical sensor. The author proposed for improvement in performance in Flow-Sensor node deployment in large size WSN.

Luo *et al.* [11] introduced a coherent framework by focusing on common problems of WSN associated to network management and versatility in networking protocols. In conventional WSN, network management is a challenging issue requires new paradigm of node retasking management through software. Manual reconfiguration of policies is done in various methods if necessary. The software-defined network (WSN) architecture that was created resembled SDN architecture. An updated version of the OpenFlow (OF) protocol was presented in a simulation model proposal. In order to analyze the effects of software-defined networking (SDN) on wireless personal area networks (WPANs) and the requirements for achieving network flexibility through the use of flow table rules and the improvement of node duty cycles in wireless networks, Costanzo et al. [12] proposed an architecture based on SDWN.

Gante *et al.* [13] proposed a smart scheme to manage WSN using SDN controller executed under base station as a new solution that is offering architecture to deal with inherited problems of energy efficiency and network management. The modified forwarding rules are introduced under network application based on SDN controller.

Kruger *et al.* [14] described a process of IoT gateway based on Raspberry Pi for LowPAN systems. The sensor nodes connected with IPv6 network to follow an application protocol under a smart water metering application.

Krylovskiy [15] discussed about use of application benchmarks for determining the virtualized layer overhead to efficient design of IoT gateways that implemented on Raspberry Pi board.

Morabito and Beijar [16] worked on multifunctional IoT gateway on Odroid C2 boards for investigating the interaction in applications of distributed data processing on hardware platforms of different parameters.

In order to support light and dense service deployment on the gateway level by Raspberry Pi board, Petrolo et al. [17] applied the virtualization concept to software using a method that involved analyzing sensor node and gateway interactions in an Internet of Things environment for applications that dynamically allocated virtualized services.

Similar type of virtualization scheme for WSN [18] proposed by Khan *et al.* [19] based on multilayer architecture for performance enhancement of WSN by enabling the sensor nodes to run concurrent tasks belonging to variety applications.

A great deal of research was done to support the viability of cloud computing and programmable network principles in WSN. However, with the exception of those by Bizanis and Kuipers [24] and Al-Kaseem and Al-Raweshidy [20], none of them included SDN and cloud computing in IoT.

Concepts centered on the use of SDN and virtualization in mobile and wireless networks were given by Yang et al. [21] and Li and Chen [22].

In their description of the development of SDN, Sood et al. [23] emphasized the optical and wireless domains that are currently integrated with SDN and IoT.

Al-Fuqaha et al.'s survey [9] is beneficial in that it offers important details on IoT-based protocols, technologies, and applications in a thorough way.

This study concentrated on the finding that sensor nodes operate autonomously, which makes network administration and control very challenging. This leads to large traffic generation, which in turn causes energy shortages and raises latency. Developing Internet of Things applications that integrate many programmable network and wireless communication protocols beneath a range of sensor gear has proven to be an enormous task. Re-tasking is necessary after node deployment, as evidenced by recent advancements in sensor network applications under Internet of Things applications. Reconfiguration is made possible and network duties are made more flexible with the help of optimization strategies.

3. Proposed Work

The proposed works is described here in two section in terms of explaining the data acquisition part in section 3.1 and QoS optimization part under IoT web service platform in section 3.2.

3.1. Thingspeak: IoT web Service

Thingspeak is an open-source, web-based API platform for the Internet of Things [4, 5, 6] that helps analyze and visualize sensed data by storing it in a graphical representation. Thingspeak communicates with connected "things" via an internet connection, which serves as a carrier for data packets. Its cloud storage integration facilitates the retrieval, storing, and analysis of data from various sensors that are linked to the host microcontroller. Thingspeak develops applications for tracking location and places, logs data from sensors, and creates a "social network" of objects and things with real-time status updates. Alternatively, it allows end users to manage automated products installed in homes, businesses, or public spaces that are linked to public domain networks online. The word "Channel" refers to Thingspeak capability, which includes data, location, and status information for connected sensor nodes. In order to receive, process, and visualize data in MATLAB and to generate responses that may be sent as tweets, SMS messages, or other alert forms, channels are constructed in "Thingspeak." Thingspeak has features that allow any registered user to create an open access channel in the public domain for tasks like analysis and estimate. The "Things/objects" are governed by analysis derived from data sensing and transmitted to PCs over the Internet; in order to accomplish this, data is uploaded to network servers known as clouds. The "Cloud" platform may be used to provide customers with virtual servers that offer graphical visualization. Most things employ sensors and actuators to communicate analog data about the state of the environment. The Internet of Things (IoT) connects everything, enables communication between objects, and lets objects engage with one another.

3.2. Grey Wolf Optimization Algorithm

Due to simple implementations, straightforward ideas and low system information requirements the metaheuristic optimization algorithms are getting very popular in modern technology. Current optimization techniques can avoid local optima and are widely applied across all fields. There exist several methods that stem from several combinatorial optimization issues. A novel method [29] called "Grey Wolf optimization" was released in 2016. The social behavior of grey wolves, which operate in a hierarchical leadership structure, serves as an inspiration for this hunting approach. The elite predators, grey wolves, often inhabit packs of five to fifteen animals. The strategy of hunting classified into four groups α, β, Δ, and Ω. α-wolves taken as leader of the group that has authority of making decision for hunting place, rest state. The α-wolves are dominant and instructing others for following them. They play a significant part in generating novel solutions. When it comes to helping α-wolves make decisions, β-wolves go above and beyond. When the alpha wolves die, they make a decision. They pay attention to the α-wolves' decisions and respond accordingly. We refer to the wolves as subordinate wolves. They are the property of scouts, elders, guardians, hunters, and sentinels. Δ-wolves oversee Ω-wolves and obey alphas and betas. wolves have the lowest position and serve as scapegoats. They trail all other wolves in charge. They don't matter since they can prevent others from having internal issues [30].

Three types of hunting are distinguished in GWO: tracking, surrounding, and assaulting the target during the exploration and exploitation phase.

Exploitation is the process of finding the best solution while surrounding and assaulting the prey, while tracking is the process of locating the best solution throughout a global search area.

When encircling, the prey's position is identified. During this stage, the prey's position vector is established, and searchers modify its location by determining the optimal solution to the equation below:

$$\vec{D} = \left|\vec{C}.\overrightarrow{X_p}(k) - \vec{X}(k)\right| \qquad (1.1)$$

$$\vec{X}(k+1) = \overrightarrow{X_p}(k) - \vec{A}.\vec{D} \qquad (1.2)$$

k: current iteration, position vector of the prey, $\vec{A}$ and $\vec{C}$: coefficient vectors, | | : absolute value, $\vec{X}$: position vector, and '·' : element-by-element multiplication. The $\vec{A}$ and $\vec{C}$ vectors are calculated by following equations:

$$\vec{A} = 2\vec{a} * \vec{r} - \vec{a} \qquad (2)$$

$$\vec{C} = 2 * \vec{r} \qquad (3)$$

$\vec{r}$: random value in [0, 1] and $\vec{a}$: decreases linearly from 2 to 0. The position of the search agent $[X, Y]$ is undergoes through adjustments on the basis of prey position obtained so far $[X^*, Y^*]$. $\vec{A}$ and $\vec{C}$ adjusted for achieving best agent in different locations.

In the phase of hunting, α-wolves direct other wolves. Initially, α-wolve is first best solution, β-wolves: 2nd best solution and Δ-wolves taken from 3rd best solution. These 3 solutions are used in update of the position of the lowest ranking solution considered under Ω-wolves. The equation followed under phase of hunting strategy are given below:

$$\vec{D}_\alpha = \left|\vec{C}_1 * \vec{X}_\alpha - \vec{X}\right| \qquad (4.1)$$

$$\vec{D}_\beta = \left|\vec{C}_2 * \vec{X}_\beta - \vec{X}\right| \qquad (4.2)$$

$$\vec{D}_\delta = \left|\vec{C}_3 * \vec{X}_\delta - \vec{X}\right| \qquad (4.3)$$

$\vec{C}_1, \vec{C}_2,$ and $\vec{C}_3$ represents coefficient vector used to adjust distance vector (eq. 3).

$\vec{D}_\alpha$, $\vec{D}_\beta$, and $\vec{D}_\delta$: modified distance vector between the α, β, and Δ-wolve position to the other wolves & $\vec{X}$ position vector of Ω-wolves.

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 * (\vec{D}_\alpha)$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 * (\vec{D}_\beta) \qquad (5)$$

$$\vec{X}_3 = \vec{X}_\Delta - \vec{A}_3 * (\vec{D}_\Delta)$$

where $\vec{X}_1$: new position vector obtained using α-wolves position $\vec{X}_\alpha$, and $\vec{D}_\alpha$, $\vec{X}_2$: new position vector obtained using β-wolves position $\vec{X}_\beta$ and $\vec{D}_\beta$, $\vec{X}_3$: new position vector from Δ-wolves position $\vec{X}_\delta$ and $\vec{D}_\beta$, and $\vec{A}_1$, $\vec{A}_2$, and $\vec{A}_3$; coefficient vectors (eq. 2).

$$\vec{X}(k+1) = \frac{\sum_{i=1}^{n} \vec{X}_i}{n} \qquad (6)$$

where $\vec{X}(K+1)_1$ is new position vector as average of all positions of α,β and Δ-wolves and n representing α,β and Δ-wolves (n is equal to three).

Attack phase is helpful in identifying local solutions. Local search performs variation in $\vec{A}_1$ in the interval [-2a to +2a]. If the value of coefficient vector $|A| < 1$ then local search is performed. With these operators, search and update of positions performed to obtain optimum value. Search of prey phase is helping in diverging each other for finding for prey and converging to attack prey. If $|\vec{A}| > 1$ then search is diverging from prey and finding the new prey.

4. Results and Discussion

The MatLab program runs the algorithm to determine the best way to minimize congestion. The program utilizes algorithms for particle swarm optimization, ant colony optimization, and grey wolf optimization to determine the optimal path for data transmission with the least amount of congestion. The recommended method improves network load balancing and prevents congestion, which increases IoT web service throughput and performance. Based on three distinct optimization techniques, the algorithm finds optimal routing pathways and secondary paths by using the principle of intelligent solutions [20, 21, 22]. By identifying and selecting the best route, the algorithm presented here generates an intelligent solution of potential fittest pathways to convey data from source point to destination. The section that follows provides a step-by-step explanation of the suggested approach:

• Detection of different paths from source node to destination node by forwarding 'Request' packet.

• Backward bounces of 'Acknowledgement' packets from the destination to source node to update routing tables of the nodes that are passed through.

• Intelligent utilization of Information of the link quality estimate from table by collected information.

• Discovering the secondary ideal paths (source node to destination node) on the moments of data is exceeding queue limits in primary path (situation of congestion).

• Backward bounces from the destination node to source node via the new paths with respect to updates in routing tables for nodes involved in secondary ideal path.

• Broadcasting of the table of estimated link quality by collection of information of local secondary links.

The algorithm proposed in this work used three data tables for in the network's node for performing routing using ideal path selection as primary ad alternative secondary path:

1. Probabilistic data structure: This table is including the values of probability of neighbor nodes that may be selected in next hop on reaching towards the destination. It is including track information for directing the data packets towards destination. It also contains route information as set of node id covered under a specific path. [23, 24, and 25]. In this manner this data structure is including following fields:

• Target-ID: It is representing the target node address.

• Next-hop-ID: It is containing next adjacent node address used for reaching the destination.

• State values field: estimated probabilities of neighboring nodes to be elected in next hop.

The probabilistic table describes for a node N containing R_N rows ($R_N = |S.N_N|$, ($S.N_N$ is set of nodes adjacent to the node N). C columns are representing total possible paths to reach the destination. For the $Pro_{L.T}$: probabilities of packets transmission through the link L. The following relation is followed for column entries as State values field tables for k^{th} device id. Hence, following relation is formulated in column of probability table:

$$\sum_{L \epsilon\, S.N_N} Pro_{L.T} = 1 \;\forall\, T \,\epsilon\, [1 \dots C] \qquad (7)$$

2. Delay data structure: This table is consisting of rate delay time value taken under passing data through the intermediate nodes. It contains N entries for each trip. The delay rates kept for M number search agents.

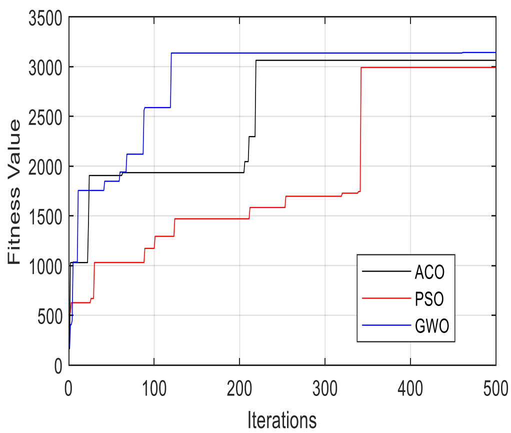

Fig 1. Fitness value convergence plot with respect to number of iterations.

3. Link estimation table: This table is containing information of quality and strength of links. The values exploit the information to measure the levels of strength of links, taking account of delay of sending packets.

Table 1: Comparative values of different optimization algorithm performance

	GWO	*ACO*	*PSO*
Iterations	120	219	342
Fitness	3137	3026	2991

Table 2: Comparative values of different optimization algorithm in terms of average delay and network set up time.

	GWO	*ACO*	*PSO*
Average Delay (sec)	0.50	0.60	0.68
Network Setup time (sec)	0.76	0.88	1.06

The MatLab software is used to write code for reading and writing data from different channels associated to ThingSpeak IoT based web services. The channels data is read iteratively from MatLab on keeping the PC connected to internet via local WiFi connections. The Channel information consists of channel id, location, number of sensors, sensors type, packet time information. The time consumed in between reception of two consecutive packets under a specific channel is considered as the delay time. In figure 1 it has been shown that three different types of optimization algorithms are integrated with code of data acquisition from Thingspeak platform on MatLab. These algorithms collect the channel information and search for the ideal primary and secondary path that gives minimum probability of delay for specific source and destination node. The ACO, PSO and GWO algorithm iteratively finds the best route that has fitness value. In this figure it has been observed that the GWO takes minimum time to achieve best fitness value [25 - 28].The summary of figure 1 is shown in table 1 in terms of fitness value that has been attained at maximum level and iteration consumed in obtained the highest fitness value. It may be observed that GWO takes minimum number of iteration and gives the optimum route with highest fitness value as compared to ACO and PSO. Figure 2 is showing the variation of delay time with respect to packet id. It may be seen that initial set up time is large but after 5 to 10 iterations the delay time is varying in range of 0 to seconds. This figure is showing comparative view the delay time at different packet id under optimum routes selected by GWO, ACO and PSO. The table 2 gives the summary in terms of average delay time consumed in receiving ant transmitting the packets and initialization set up time. It may be observed that the GWO takes minimum set up time and average delay time as compared to ACO and PSO.

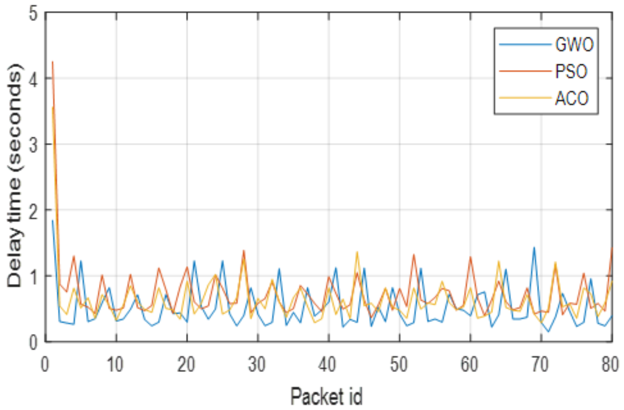

Fig 2. Delay time variation with respect to packet id.

Figure 3 is demonstrating the results as the bar chart for demonstration of latency in seconds for reading data from Thingspeak to user from channel id 12397,234684 and 1293177 and sending (writing) data from user to ThingSpeak to channel id 17504.The reading and writing latency in seconds is recorded at different packet id arrival or transmission through Thingspeak-Matlab interaction. Total latency on reading or writing mode using optimized route selection through GWO, ACO and PSO is shown in table 3 and figure 4.It has been observed from figure 4 that GWO gives minimum latency for all channels during reading or writing mode. Due to lower latency, less delay and fast convergence GWO is capable of giving highest throughput (Table 3).

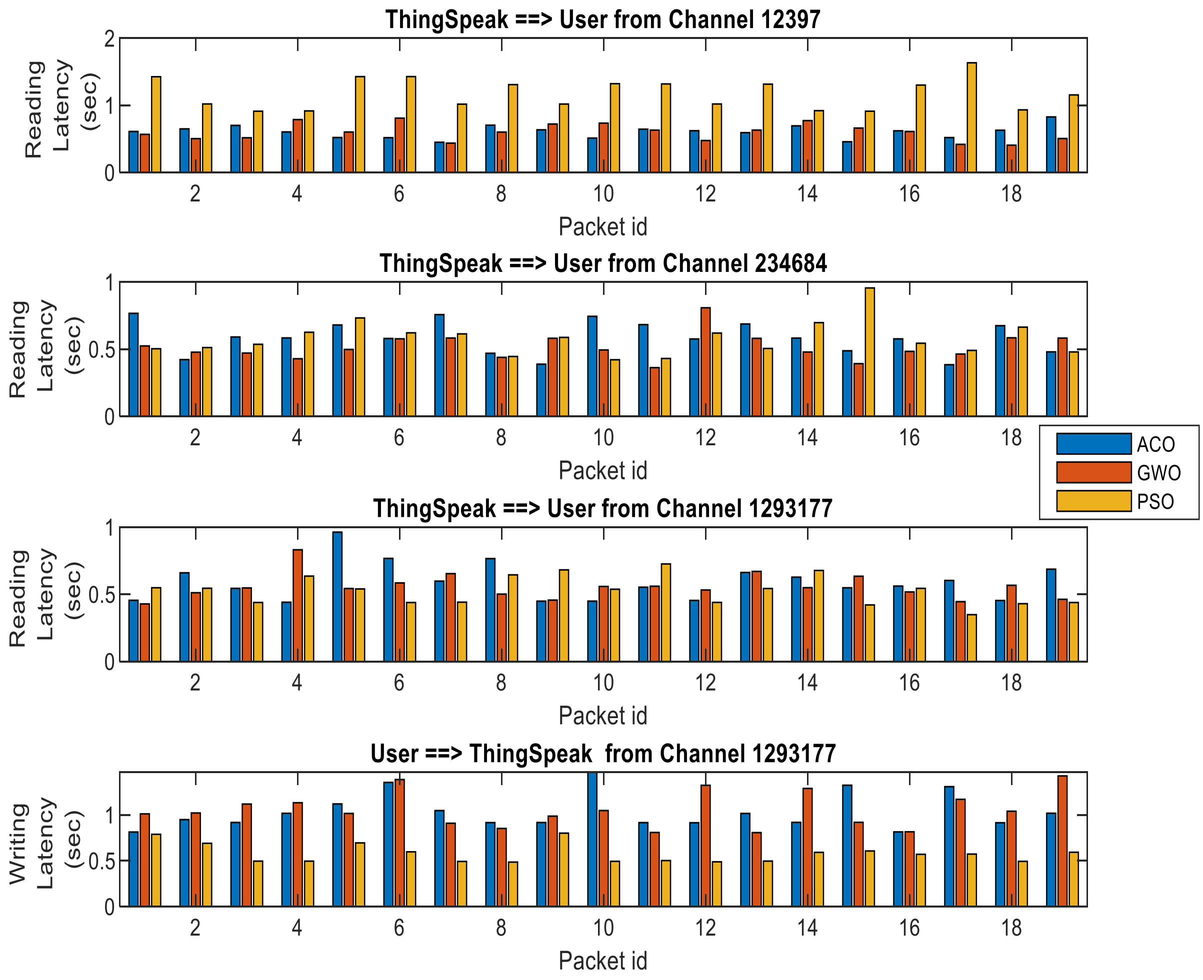

Figure 3: Reading and writing latency for user and Thingspeak channels using GWO, ACO and PSO.

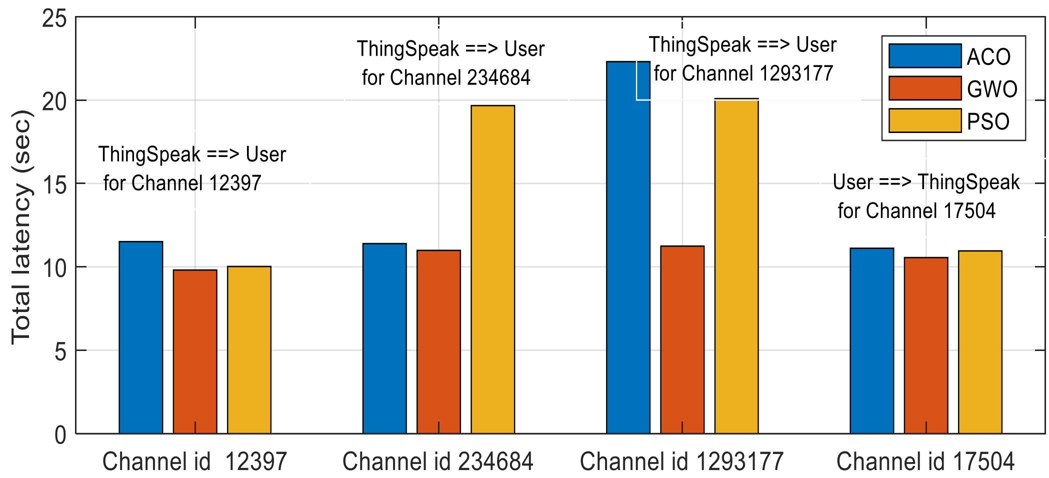

Figure 4: Total latency at different channels using GWO, ACO and PSO.

Table 3: Comparison of PSO, GWO and ACO performance in terms of total latency on reading and writing mode through different channels

	Total Latency (sec)	***PSO***	***GWO***	***ACO***
Reading mode	**ThingSpeak to user for channel 12397**	11.51	9.80	10.02
	ThingSpeak to user for channel 234684	11.39	10.98	19.67
	ThingSpeak to user for channel 1293177	22.30	11.24	20.08
	Total Reading Mode Latency (sec)	45.20	32.03	49.77
Write Mode	**User to ThingSpeak for channel 17504**	11.11	10.95	10.55
	Throughput (Kbps)	3200	3765	3025

5. Conclusions

By balancing a load of data packets with other secondary ideal channels, this study suggested an effective routing method for IoT base web services that identifies and avoids congestion problems in the routing. The suggested method is predicated on its fundamental functions, which employ several optimization techniques. Three fundamental processes underpin the suggested work: identifying severe congestion on the best data transmission pathways, generating backup paths using the most recent search parameter values, and distributing traffic loads across primary and secondary routes. The ThingSpeak based IoT web service integration to MatLab software is used to justify the effectiveness by evaluation of performance with GWO, PSO and ACO. Simulation results are demonstrating the better performance by improving throughput, end-to-end delay rate, and latency using GWO compared to other approaches. The results are showing that the GWO gives better throughput compared to PSO and ACO algorithms. The results are also showing that better average end-to-end delay is achieved.

As a future scope, development is suggested in terms of associating effective mechanisms of identifying numerous paths on consideration of appropriate channel space utilization and multiple radios intra/inters flow interference. The performance of proposed approach compared to other intelligent algorithms followed by other performance parameters. Add on of mechanisms for enhancing security for avoiding network attacks that affect network performance negatively.

Acknowledgment

Manuscript Communication Number (MCN): IU/R&D/2024-MCN-0002749 Office of Research and Development, Integral University, Lucknow.

This research was supported by Integral University. We thank our colleagues who provided insight and expertise that greatly assisted the research.

Author contributions

Saima Aleem[1]: Writing-Original draft preparation, Software, **Shish Ahmad[2]:** Conceptualization, Methodology.

References

[1] D. Meisner, C. M. Sadler, L. A. Barroso, W. D. Weber and T. F. Wenisch, "Power management of online data-intensive services," Computer Architecture (ISCA), 2011 38th Annual International Symposium on, San Jose, CA, 2011, pp. 319-330.

[2] Anurag, R. S. (2020). Load Forecasting by using ANFIS. International Journal of Research and Development in Applied Science and Engineering, 20(1).

[3] “Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021,” San Jose, CA, USA, Cisco, White Paper, 2017. Accessed: Oct. 2018.

[4] N. C. Luong *et al.*, “Data collection and wireless communication in Internet of Things (IoT) using economic analysis and pricing models: A survey,” *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2546–2590, 4th Quart., 2016.

[5] V. Gazis, “A survey of standards for machine-to-machine and the Internet of Things,” *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 482–511, 1st Quart., 2017.

[6] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng,“IoT middleware: A survey on issues and enabling technologies,” IEEE Internet Things J., vol. 4, no. 1, pp. 1–20, Feb. 2017.

[7] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, “A survey on 5G networks for the Internet of Things: Communication technologies and challenges,” *IEEE Access*, vol. 6, pp. 3619–3647, 2018.

[8] A. Aijaz and A. H. Aghvami, “Cognitive machine-to-machine communications for Internet-of-Things: A protocol stack perspective,” IEEE Internet Things J., vol. 2, no. 2, pp. 103–112, Apr. 2015.

[9] Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A survey on enabling technologies, protocols, and

applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[10] Mahmud, R. Rahmani, and T. Kanter, "Deployment of flow-sensors in Internet of Things' virtualization via OpenFlow," in *Proc. 3rd FTRA Int. Conf. Mobile Ubiquitous Intell. Comput. (MUSIC)*, Jun. 2012, pp. 195–200.

[11] T. Luo, H.-P. Tan, and T. Q. S. Quek, "Sensor OpenFlow: Enabling software-defined wireless sensor networks," *IEEE Commun. Lett.*, vol. 16, no. 11, pp. 1896–1899, Nov. 2012.

[12] S. Costanzo, L. Galluccio, G. Morabito, and S. Palazzo, "Software defined wireless networks: Unbridling SDNs," in *Proc. Eur. Workshop Softw. Defined Netw.*, Oct. 2012, pp. 1–6.

[13] A. D. Gante, M. Aslan, and A. Matrawy, "Smart wireless sensor network management based on software-defined networking," in Proc. 27th Biennial Symp. Commun. (QBSC), Jun. 2014, pp. 71–75.

[14] A. P. Kruger, A. M. Abu-Mahfouz, and G. P. Hancke, "Rapid prototyping of a wireless sensor network gateway for the Internet of Things using off-the-shelf components," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Mar. 2015, pp. 1926–1931.

[15] Krylovskiy, "Internet of Things gateways meet Linux containers: Performance evaluation and discussion," in *Proc. IEEE 2nd World Forum Internet Things (WF-IoT)*, Dec. 2015, pp. 222–227.

[16] R. Morabito and N. Beijar, "Enabling data processing at the network edge through lightweight virtualization technologies," in *Proc. IEEE Int. Conf. Sens. Commun. Netw. (SECON Workshops)*, London, U.K., Jun. 2016, pp. 1–6.

[17] R. Petrolo, R. Morabito, V. Loscrì, and N. Mitton, "The design of the gateway for the Cloud of Things," *Ann. Telecommun.*, vol. 72, nos. 1–2, pp. 31–40, 2017, doi: 10.1007/s12243-016-0521-z.

[18] Khan *et al.*, "Wireless sensor network virtualization: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 553–576, 1st Quart., 2016.

[19] Khan, F. Belqasmi, R. Glitho, and N. Crespi, "A multi-layer architecture for wireless sensor network virtualization," in *Proc. 6th Joint IFIP Wireless Mobile Netw. Conf. (WMNC)*, Apr. 2013, pp. 1–4.

[20] R. Al-Kaseem and H. S. Al-Raweshidy, "SD-NFV as an energy efficient approach for M2M networks using cloud-based 6LoWPAN testbed," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1787–1797, Oct. 2017.

[21] M. Yang *et al.*, "Software-defined and virtualized future mobile and wireless networks: A survey," *Mobile Netw. Appl.*, vol. 20, no. 1, pp. 4–18, 2015, doi: 10.1007/s11036-014-0533-8.

[22] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[23] Sood, S. Yu, and Y. Xiang, "Software-defined wireless networking opportunities and challenges for Internet-of-Things: A review," *IEEE Internet Things J.*, vol. 3, no. 4, pp. 453–463, Aug. 2016.

[24] N. Bizanis and F. A. Kuipers, "SDN and virtualization solutions for the Internet of Things: A survey," *IEEE Access*, vol. 4, pp. 5591–5606, 2016.

[25] Anurag, R. S. (2020). Modern Trends on Image Segmentation for Data Analysis-A Review. International Journal of Research and Development in Applied Science and Engineering, 20(1).

[26] Yadav, M. S., & Ahamad, S. (2020). Outlier detection in WSN by entropy based machine learning approach. Indonesian Journal of Electrical Engineering and Computer Science, 20(3), 1435-1443.

[27] Khan, W., & Haroon, M. (2022). An unsupervised deep learning ensemble model for anomaly detection in static attributed social networks. International Journal of Cognitive Computing in Engineering, 3, 153-160.

[28] Fatima, S., & Ahmad, S. (2020). Secure and effective key management using secret sharing schemes in cloud computing. International Journal of e-Collaboration (IJeC), 16(1), 1-15.

[29] Haroon, M., Misra, D. K., Husain, M., Tripathi, M. M., & Khan, A. (2023). Security issues in the internet of things for the development of smart cities. In Advances in Cyberology and the Advent of the Next-Gen Information Revolution (pp. 123-137). IGI Global.

[30] Aleem, S., & Ahmed, S. (2023). A review of the security architecture for SDN in light of its security issues. *International Journal of Scientific Research in Network Security and Communication, 11*(3), 8-14.