International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2023/218512 Volume 10, Issue 1, January – February (2023)

RESEARCH ARTICLE

Ornstein Uhlenbeck Cache Obliviousness Neural Congestion Control in Wireless Network for IOT Data Transmission

N Thrimoorthy
School of Computer Science and Applications, REVA University, Bangalore, India.
thrimoorthy.n@reva.edu.in

Somashekhara Reddy D
Department of Computer Science and Engineering, Jain University, Bangalore, India.
somashekhara.mtech@gmail.com

Chandramma R
Department of Computer Science and Engineering, Jain University, Bangalore, India.
Chandramma.cse@gmail.com

Soumya Unnikrishnan
Department of Computer Science, St. Francis College, Bangalore, India.
soumya.u@stfranciscollege.edu.in

Vanitha K
Department of Computer Science and Engineering, Jain University, Bangalore, India.
profvanithacse@gmail.com

Received: 06 December 2022 / Revised: 18 January 2023 / Accepted: 21 January 2023 / Published: 26 February 2023

Abstract – Wireless Network is one of the Internet-of-Things (IoT) prototypes that come up with monitoring services, therefore, influencing the life of human beings. To ensure efficiency and robustness, Quality-of-Service (QoS) is of the predominant point at issue. Congestion in wireless networks will moreover minimize the anticipated QoS of the related applications. Motivated by this, a novel method called, Ornstein–Uhlenbeck Transition and Cache Obliviousness Neural Adaptive (OUT-CONA) to improve congestion control of wireless mesh networks is presented. Adaptive actor-critic deep reinforcement learning scheme on Ornstein–Uhlenbeck State Transition scheduling model to address handovers during data transmission for IoT-enabled Wireless Networks is first designed. Here, by employing the Ornstein–Uhlenbeck state transition scheduling model, both the advantages of the Gauss and Markov Processes are exploited, therefore reducing the energy consumption involved while performing the transition. Next, in the OUT-CONA method, LSTM is imposed for learning the current state representation. The LSTM with the current state representation achieves the objective of controlling congestion with cache obliviousness. The Cache Obliviousness-based Congestion method is utilized for congestion control with obliviousness caching using coherent shielding among organized as well as disorganized data. Furthermore, the performance of the OUT-CONA method is evaluated and compares the results with the performances of conventional techniques, adaptive aggregation as well as hybrid deep learning. The evaluation of the OUT-CONA congestion control method attains better network using lesser misclassification rate, consumption of energy, delay as well as higher goodput using conventional methods in Wireless Mesh Networks.

Index Terms – Wireless Mesh Network, Internet of Things, Ornstein–Uhlenbeck, Transition, Cache Obliviousness, Neural Adaptive, Congestion Control.

1. INTRODUCTION

In the present era of advancement, communication is taken as the most paramount requirement. Moreover, the notable surge necessitates better bandwidth, optimal resource distribution, as well as services in need for realizing the fundamental requirements. One of the distinguished examples is the swift advancement of technology transformation from 2G to 4G as well as the forthcoming 5G and 6G. Among the several challenges, systematic congestion control is contemplated as one of the pivotal factors that permit the operators in

RESEARCH ARTICLE

executing numerous network instances for a better quality of services.

An adaptive aggregation method was proposed by [1] based on the characteristics of IoT traffic to reduce network traffic congestion. Here, the aggregation model was applied to payload exchange as well as traffic networks for utilizing periodic networks, with which the performance of competence was said to be optimized. With the application of the adaptive aggregation model, the large amount of data was said to be reduced and also minimized the collision probability even in the case of a dense number of nodes and finally resulting in less traffic congestion. Due to this, the throughput along with collision probability was improved therefore reducing the network congestion

Despite improvement observed in throughput and collision probability, the energy consumed while analyzing network traffic congestion towards data transmission was not focused. To address this issue, an Ornstein–Uhlenbeck State Energy-efficient Transition scheduling employing Gauss and Markov Processes is utilized that with optimal transitions between sensors or devices not only minimizes energy consumption but also reduces the misclassification rate by addressing the handover.

A deep learning method was presented in [2] consisting of LSTM as well as SVM. Here, a smart decision-making technique was designed so that load balancing was ensured for the incoming network traffic. Also, network slice failure was restricted with alternative slices provided in case of failure or during overload scenarios. Due to this, hybrid designing, accuracy was improved with minimum time consumption and misclassification rate.

Despite improvement observed in terms of reducing the time consumption with optimal load balancing for incoming network traffic and hence avoiding congestion, the delay involved during the overall analysis was not focused, therefore compromising the rate of goodput. To address this issue and handle handovers for congestion control in the wireless network for IoT data transmission, the Cache Obliviousness-based Neural Adaptive Congestion Control model is designed. With the design of this Cache Obliviousness-based Neural Adaptive Congestion Control model goodput is improved using lesser delay.

Despite IoT transforming unparalleled momentum, additional contemporary studies, as well as growth, concentrate on IoT transforming in connection with the regulation of enormous sensors. In analogous with this direction, enormous machine-type communications to smooth transmission is the vision of fundamentals utilization to 5G mobile as well as wireless networks. Interrelation of devices in a group of issues, like efficient data collection, congestion, reliable transmission, and so on.

Three novel mechanisms were proposed in [3] to ensure access according to the priority. On the basis of these three 5G new radio frame, OUT-CONA method enabled the devices on basis of device vicinity and distributes exclusive prologues for grouped devices for random access. With this, grant-free transmission was ensured. Despite improvement observed in managing the load, some side effects were found to be compromised like higher communication costs, delay, PLR, throughput etc.

Dynamic Hop Selection Static Routing Protocol was designed in [4] to address load balancing issues with the selection of route paths within congestion. However, the above decision-making processes frequently necessitate enormous data transmissions between sensors, that in turn necessitates data as well as correct data analyses performed on the basis of huge traffic data.

Taking into consideration, large data-driven, as well as the nonparametric method, was proposed in [5] that obtained traffic outline based on time to precise as well as effective traffic flow prediction, therefore improving the prediction accuracy. However, according to the network situation transmission rate has to be managed. To focus on this aspect, a novel congestion control method was designed in [6] for ensuring high throughput. The designed meth failed to consider the smart city monitoring systems, and healthcare applications.

Yet another method to reduce network power consumption while handling congestion was designed in [7]. Requirements for steady and effective data transmission are elevating owing to the continuing evolution of wireless data traffic globally. An advanced congestion control mechanism employed different margins by CoAP. In [8], a queuing delay was designed with the purpose of solving the utility maximization issues, via perfect scheduling and distributed scheduling. However, the network performance was not achieved by queuing delay.

1.1. Problem Statement

In IoT, dissimilar devices were focused on necessities such as communication speed, delay, and reliability. But, most of the IoT devices failed to consider the delay metric. Thus, data transmission is a challenging factor in communication. Traffic congestion control is a demanding task in 5G/6G wireless network technologies. It is a dangerous issue for mobile operators and commercial businesses for the next generation of communication technology. Despite significant advantages provided by WMN, its practical distribution to connect Internet of Things (IoT) networks caused very expensive congestion. Conventional forecasts as well as classification methods for controlling congestion for IoT-enabled wireless networks will not provide worthwhile achievement as well as precise outcomes.

RESEARCH ARTICLE

To address the issue, in this work, a mechanism for accurate and robust controlling of congestion is developed. In IoT-enabled WMN sensor-enabled network scenarios within IoT, the WMN technique uses radio sensors for the transmission of data. To reduce energy consumption and misclassification rate during data transition, the technique also calculate all sensors by Ornstein–Uhlenbeck State Transition and achieves transitions. Next, the Cache Obliviousness-based Neural Adaptive Congestion Control model is employed for controlling the congestion arising during data transmission

1.2. Contributions

The proposed work contributions are listed below.

- Ornstein–Uhlenbeck State Transition is used to classify node status of IoT within administering strategies: Operative 'O', Idle 'I' and 'B' to flexible radio node scheduling scheme. It was a well-defined administering strategy that recognizes nodes possessing residual energy with minimum misclassification rate, particularly in idle mode. Node scheduling scheme was utilized in OUT-CONA of data packets among nodes are transferred is stimulated.
- To develop a Cache Obliviousness Neural Adaptive Congestion Control model to control congestion using cache obliviousness ensuring coherent shielding between the organized and disorganized flow of data packets. Then, congestion incurred is minimized using improved goodput as well as d delay.
- Verifying the effectiveness of OUT-CONA in terms of energy consumption, misclassification rate, goodput as well as delay compared with conventional methods.

1.3. Organization of the Paper

The article is summarized by. Section 2 reviews explain IoT enabled congestion control method for WMN the proposed Ornstein–Uhlenbeck Transition and Cache Obliviousness Neural Adaptive (OUT-CONA) using wireless network system model, IoT enabled Data Communication Wireless Mesh Network scheme for providing congestion control model presented in Section 3. Section 4 describes the experimental assessment of proposed and existing methods. Section 5 explains the conclusion of the article.

2. RELATED WORK

With the inception of the third generation of mobile cellular and wireless networks, primary phase of wireless access is considered. However, it remains a certain amount of issues owing to the distinct nature and essentials of wireless networks. The motivation of congestion control is to make sure network stability and accomplish a considerately trustworthy administering of the network resources between the users.

In [9], a novel traffic control method on the basis of data offloading method was designed by employing VCG as well as the Rubinstein bargaining game scheme. With these data offloading mechanism, IoT traffic congestion was avoided, and also enhanced the quality-of-service (QoS) accordingly. But, the delay was not reduced.

Traffic prediction is predominant to make certain perpetual system competence and establish the quality of service for IoT, heavily depending on congestion, bandwidth allocation, and so on. In [10], an elaborate outline of the IoT traffic method employing time series as well as ANN was developed and the purpose of the prediction network was also utilized. The persistent data connections as well as transmissions of data were deliberately administered within prevailing networks, causing considerable issues for wireless networks. Also, the design algorithm failed to consider all dynamic parameters of the IoT environment.

In [11], the traffic scheduling issue was addressed and also hybrid routing forwarding model along with the congestion control algorithm was designed to accomplish a practical solution. Here, the assignment of traffic was initially transformed into multi knapsack problem, followed by which Artificial Fish Swarm Algorithm (AFSA) was employed to ensure congestion control. Owing to the extensive data in materializing IoT, controlling congestion of traffic flow was essential for attaining explicit protection and QoS. However, in the event of a congested network, information is highly susceptible to packet drop, therefore, degrading the entire network.

A Proportional Integrator Differentiator was developed by [12] congestion control rate, therefore, ensuring stability and packet delivery. However dynamic traffic characteristics still face issues while controlling congestion like high throughput and collision probability. To address these aspects, traffic aggregation techniques for IoT network was designed in [13] for investigating traffic parameters. Despite improvement observed in both throughput and collision measures, with a considerable amount of different data flows the computational cost in turn increased. To solve this issue, machine learning classification methods in [14] for predicting traffic control and accordingly congestion were also avoided based on differentiation mechanisms. Different types of sensors were integrated into automated and smart systems with the purpose of sensing, collecting, and transferring the data. But, computational complexity was higher.

One of the emerging technologies that enhance both the observation and potentialities of smart systems is machine learning. In [15], an IoT enabled intelligent traffic congestion handling scheme was designed specifically for smart societies. The proposed method also sensed and notified the probable areas of congestion. But, Time delays were a vital issue and occur owing to poor collection of sensors or slow

RESEARCH ARTICLE

connectivity of the network. Yet another method using a clustering mechanism was proposed in [16] employing a hierarchical mechanism. With this type of mechanism, both consumptions of energy and delay were reduced drastically. The designed mechanism failed to minimize the misclassification rate.

CACC method specifically designed for lightweight constrained application protocol (CoAP) was proposed in [17]. The congestion control method here was designed by conventional mechanism recommended lightweight interface was also designed. With these design mechanisms, not only congestion was controlled but also retransmission was avoided to a greater extent. However, the energy involved in controlling the congestion was not focused.

Path determination architecture was developed in [18] to achieve superior performance in terms of both energy and extending network lifetime by utilizing exponential smoothing. But, it failed to consider the high volume of data transmission. A game theory mechanism was introduced in [19] to mitigate the congestion by fine-tuning the sending rate. The packet loss rate was said to be minimized as well as network performance was said to be improved. However, the goodput was not measured. Improving the Congestion control mechanism of TCP was investigated in [20] to ensure throughput and minimize retransmission. But, the consumption of energy was not reduced.

2.1. Issues and Challenges

IoT smart city networks undergo difficult IoT traffic individuality by unstructured data types. Numerous aggregation methods have been developed with higher performance factors in smart city networks. The issues and challenges such as minimum throughput, maximum collision probability, a huge number of terminal devices by a varied number of transmissions at smart city traffic types, serious data traffic in several cases as the payload traffic type, maximum traffic congestion, as well as maximum overheads are considered in IoT data transmission. To overcome the issue, a deep neural adaptive method that is needed in IoT to control congestion towards data transmission in a wireless network with higher goodput and lesser energy consumption.

3. ORNSTEIN–UHLENBECK TRANSITION AND CACHE OBLIVIOUSNESS NEURAL ADAPTIVE (OUT-CONA) CONGESTION CONTROL FOR INTERNET OF THINGS

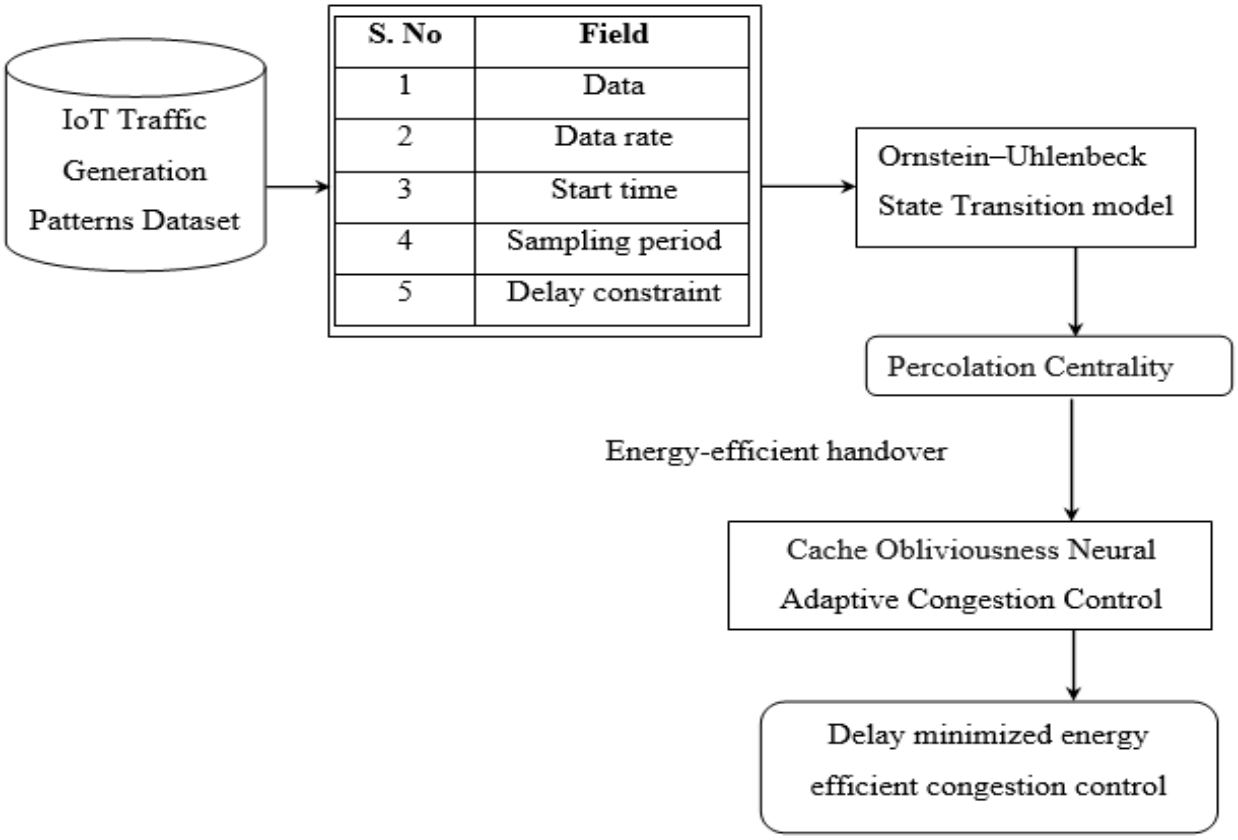

Figure 1 Block Diagram of Ornstein–Uhlenbeck Transition and Cache Obliviousness Neural Adaptive (OUT-CONA) Method

RESEARCH ARTICLE

IoT consists of both confined and non-confined devices. These confined types of devices or nodes include power, processing of devices, memory-confined nature, and so on. To implement data transmission for IoT in WMN well in a confined environment, a congestion control mechanism must inspect these confinements. This is owing to the reason that the congestion in an IoT network during data transmission results brings about a deterioration in its quality of service (QoS). Also while performing data transmission, the occurrences or handover

(i.e., data session transferred from one state to another without disconnecting the session) has to be handled. To address these aspects, an Ornstein–Uhlenbeck Transition and Cache Obliviousness Neural Adaptive (OUT-CONA) IoT-enabled congestion control for efficient data transmission is developed. The detailed explanation of OUT-CONA is explained. Block diagram of Ornstein–Uhlenbeck Transition and Cache Obliviousness Neural Adaptive (OUT-CONA) method is shown in Figure 1.

As given in the above section, the raw sensor readings collected in the laboratory from sensors obtained as input are provided to the Ornstein–Uhlenbeck State Transition model. Here, a smooth transition between different incoming data packets arriving from the host computer or the source node is modeled. With this, a smooth transition is ensured therefore addressing handover by means of the Percolation Centrality function. Second, a Cache Obliviousness Neural Adaptive Congestion Control model for smooth and fine-grained IoT data transmission between nodes or devices is provided via LSTM with a Cache Obliviousness function.

3.1. Overview of Wireless Network System Model

Congestion control is said to occur in the network layer due to an increase in the traffic generation pattern of each device in the network. With the increase in delay, the performance is said to be reduced, resulting in maximum handovers and therefore congestion, causing circumstances unpleasant. In this section, an IoT-enabled Data Communication Wireless Mesh Network model is first designed. Followed by which a detailed description of the method, Ornstein–Uhlenbeck Transition and Cache Obliviousness Neural Adaptive (OUT-CONA) is provided.

3.2. IoT Enabled Data Communication Wireless Mesh Network Model

Wireless Mesh Network (WMN) comprises several radio nodes '$N = \{N_1, N_2, \dots, N_n\}$' designed in the form of mesh topology that dispenses data packets '$DP = \{DP_1, DP_2, \dots, DP_n\}$' between devices or radio nodes. The design of the OUT-CONA method is based on the diversified-control path algorithm. The network of OUT-CONA method mesh routers is modeled via a Directed Graph '$DG = (V, E)$'. Here, 'V' denotes the set of radio nodes (i.e. nodes) '$N = \{N_1, N_2, \dots, N_n\}$' and 'E' denotes the pair of links '(i, j)' between two radio nodes 'N_i' and 'N_j'. The connection and distribution of data packets in our work for IoT-enabled data transmission in WMN between several radio nodes are established via a routing table. The content of the routing table is provided in table 1.

Table 1 WMN Routing Table Information

S. No	Fields	Units
1	Data 'D'	(KB)
2	Data rate 'DR'	(kbps)
3	Start time 'ST'	(second)
4	Sampling period 'SP'	(second)
5	Delay constraint 'DC'	(second)

As provided in the above routing table, the raw sensor readings obtained from the IoT traffic generation patterns dataset, the actual data 'D', the data rate 'DR', start time 'ST', sampling period 'SP' and delay constraint 'DC' respectively are provided as input to the state transition model in the form of data packets for each node. Using the above contents in the routing table, an IoT-enabled congestion control method on the basis of the below system model is structured.

3.3. Ornstein–Uhlenbeck State Energy-Efficient Transition Model

With the swift growth of IoT devices more traffic generation is said to occur and therefore causes a stage of congestion on the Internet. Hence, a framework that ensures a congestion control mechanism to handle all demands of distinct devices for communication is required. In this section, the first Ornstein–Uhlenbeck State Energy-efficient Transition model to differentiate states of incoming data packets from the host computer or the source node is designed in WMN for IoT data transmission.

With the deployment of the Ornstein–Uhlenbeck state transition, both the advantages of the Gauss Process and Markov Process are exploited, therefore reducing the energy consumption involved while performing the transition. In other words, the incoming data packets from the host computer with a stochastic pattern are based not only on the sequence of possible arrival rate at which the probability of incoming data packets depends on the state attained in the previous event (i.e., Markov Process) but also every finite collection of those random incoming data packets has a multivariate normal distribution (i.e., Gauss Process). Also, handover between devices is achieved by applying the Percolation Centrality.

International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2023/218512 Volume 10, Issue 1, January – February (2023)

RESEARCH ARTICLE

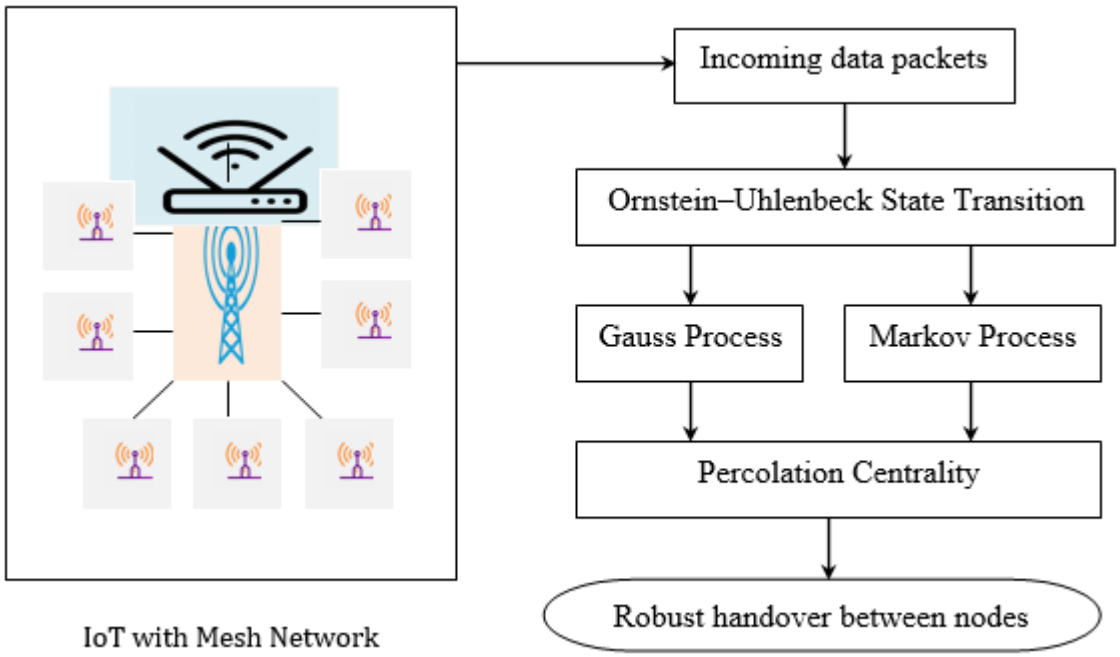

Figure 2 Block Diagram of Ornstein–Uhlenbeck State Energy-Efficient Transition Model

From the above figure 2, various radio nodes or nodes '$N = \{N_1, N_2, \dots, N_n\}$', are connected to the router 'R' via IoT gateway node 'GN' for IoT-enabled congestion control in WMN for distinct data packets '$DP = \{DP_1, DP_2, \dots, DP_n\}$'. The structuring of a node is designed in a clockwise manner, that a node possessing maximum or higher residual energy only has the potential to transfer to a lower residual energy node and not in an anticlockwise manner.

Specifically, the Ornstein–Uhlenbeck State Energy-efficient Transition along with the Gauss and Markov processes are utilized in obtaining the probabilistic node state transitions based on the constraints of the available data packets in the routing table. Moreover, to minimize the congestion due to many IoT nodes in WMN, additional optimization of administering strategies is stopped upon the detection of the idle mode. In this manner, network effectiveness is attained in terms of both energy consumption and misclassification rate in IoT-enabled data communication.

The Ornstein–Uhlenbeck process for the corresponding data packet 'DP_t' to be transmitted between nodes or the arrival of data packets is mathematically represented by the following equation (1).

$$dDP_t = -\theta DP_t dt + \sigma dWt \qquad (1)$$

From the above equation (1), the data packet 'DP_t' of the respective node contains the information provided in the routing table in addition to the continuous time stochastic arrival of data packets 'dWt' respectively. However, with the random node positioning and topological changes observed in the network owing to different numbers of IoT sensors or nodes ready for transmission, the above continuous time stochastic arrival of data packets is modified and mathematically stated as given below.

$$\frac{dDP_t}{dt} = -\theta DP_t + \sigma\eta(t) \qquad (2)$$

From the above equation (2), '$\eta(t)$' represent the noise factor (with a frequency range of 20 to 20000 hertz, depending on the arrival rate of data packets) involved during the arrival of data packets respectively. With the above Ornstein–Uhlenbeck function, Gauss and Markov processes are utilized in obtaining the probabilistic node state transitions Gauss and Markov's processes are formulated for obtaining probabilistic node state transitions so that upon detection of idle mode, with the aid of Percolation Centrality function, data session transferred from one state to another without disconnecting the session, therefore reducing the residual energy. This robust handover is said to be attained therefore reducing the probability of misclassification rate also.

The Gauss process with multiple approximations for different numbers of data packets obtained from distinct nodes (i.e., possessing different energy) in case of operative mode retains good accuracy, therefore, minimizing the misclassification rate while drastically reducing energy consumption rate. The time-continuous stochastic arrival of data packets is said to be

RESEARCH ARTICLE

Gaussian if the indices or the collection of data packets '$t_1, t_2, \dots, t_n$' in the index set 'IS' given as below.

$$DP_{t_1,t_2,\dots,t_k} = \left(DP_{t_1}, DP_{t_2}, \dots, DP_{t_k}\right) \quad (3)$$

From the above equation (3), On the other hand, the Markov process describes a sequence of probable data packets in the queue in which the probability of each arrival of a data packet depends only on the state attained in the arrival process in case of broadcasting mode. This is mathematically stated as given below.

$$Prob\ (DP_{n+1} = dp|\ DP_1 = dp_1, DP_2 = dp_2, \dots\dots, DP_n = dp_n) \quad (4)$$

From the above equation (4), the possible values of 'DP_n' forms the cardinality set referred to as the state space model. Finally, the Percolation Centrality function considers the source and target node state of each shortest route in estimating the weight. The Percolation Centrality function in our work is utilized as there arises a probability of changing the source and target node state as the data packets in the WMN propagates very fast, therefore compromising the classification rate. With the objective of reducing the misclassification rate Percolation Centrality function is employed in the proposed method. With the aid of this function, whether percolation is occurred or not are said to be obtained therefore reducing the misclassification rate.

$$PC_t(v) = \frac{1}{n}\sum_{i\neq j\neq v}\frac{\alpha_{ij}(v)}{\alpha_{ij}}\frac{DP_t^T}{\sum[DP_i^T]-DP_v^T} \quad (5)$$

From the above equation (5), the result of the Percolation Centrality function '$PC_t(v)$' is arrived at based on the total shortest routes between node 'i' and 'j', 'α_{ij}', number of those routes that traverse through 'v', '$\alpha_{ij}(v)$', the percolation time of node 'i' at time 'T' 'DP_i^T'. The pseudo-code representation of Ornstein–Uhlenbeck State Energy-efficient Transition is given in Algorithm 1.

Input: Dataset 'DS', nodes '$N = \{N_1, N_2, \dots, N_n\}$', data packets '$DP = \{DP_1, DP_2, \dots, DP_n\}$'

Output: Energy-efficient data transition

Step 1: Initialize actual data 'D', the data rate 'DR', start time 'ST', sampling period 'SP' and delay constraint 'DC', time 'T'

Step 2: Begin

Step 3: For each Dataset 'DS' with nodes 'N' and data packets 'DP'

Step 4: Obtain arrival of data packets by utilizing Ornstein–Uhlenbeck function as in equation (1)

Step 5: Measure continuous time stochastic arrival of data packets as in equation (2)

Step 6: Estimate Gauss process in case of operative mode as in equation (3)

Step 7: Estimate Markov process in case of broadcasting mode as in equation (4)

Step 8: Evaluate Percolation Centrality function as in equation (5) for handovers between operative mode and broadcasting mode

Step 9: Return energy-efficient data packet transitions

Step 10: End for

Step 11: End

Algorithm1 Ornstein–Uhlenbeck State Energy-Efficient Transition

As given in the above (Algorithm 1) Ornstein–Uhlenbeck State Energy-efficient Transition algorithm, the objective remains in reducing the energy consumption and misclassification during transition according to the three distinct administering strategies for incoming data packets. With these objectives first, Ornstein–Uhlenbeck function is employed for analyzing various incoming data packets. Second, for operative and broadcasting modes, two distinct Gauss and Markov processes are employed separately, therefore reducing energy consumption owing to the avoidance of idle mode during operation. Finally, the Percolation Centrality function is applied to handle handovers between operative and broadcasting modes, therefore, reducing the misclassification rate involved where data sessions transferring between states without explicit disconnection.

3.4. Cache Obliviousness Neural Adaptive Congestion Control Data Transmission Model

With the optimized State Transition results, a robust congestion control model for smooth and fine-grained IoT data transmission has to be designed. In this section, LSTM is imposed for learning the current state representation. The LSTM with the current state representation achieves the objective of congestion control by means of the Cache Obliviousness function. The Cache Obliviousness function here takes the advantage of a processor cache without assigning the size of the cache as a definite parameter.

The Cache Obliviousness-based Congestion mechanism is applied with the purpose of controlling congestion by means of obliviousness caching with coherent shielding between the organized and disorganized flow of data packets. In this manner, a robust congestion control with minimum end-to-end delay and maximum goodput between nodes or devices is said to be ensured. The Cache Obliviousness Neural Adaptive Congestion Control Data Transmission model is composed of four layers. They are one input layer, one hidden layer, one

RESEARCH ARTICLE

quantization layer, and the output layer respectively. Figure 3 shows the block diagram of the Cache Obliviousness Neural Adaptive Congestion Control Data Transmission model.

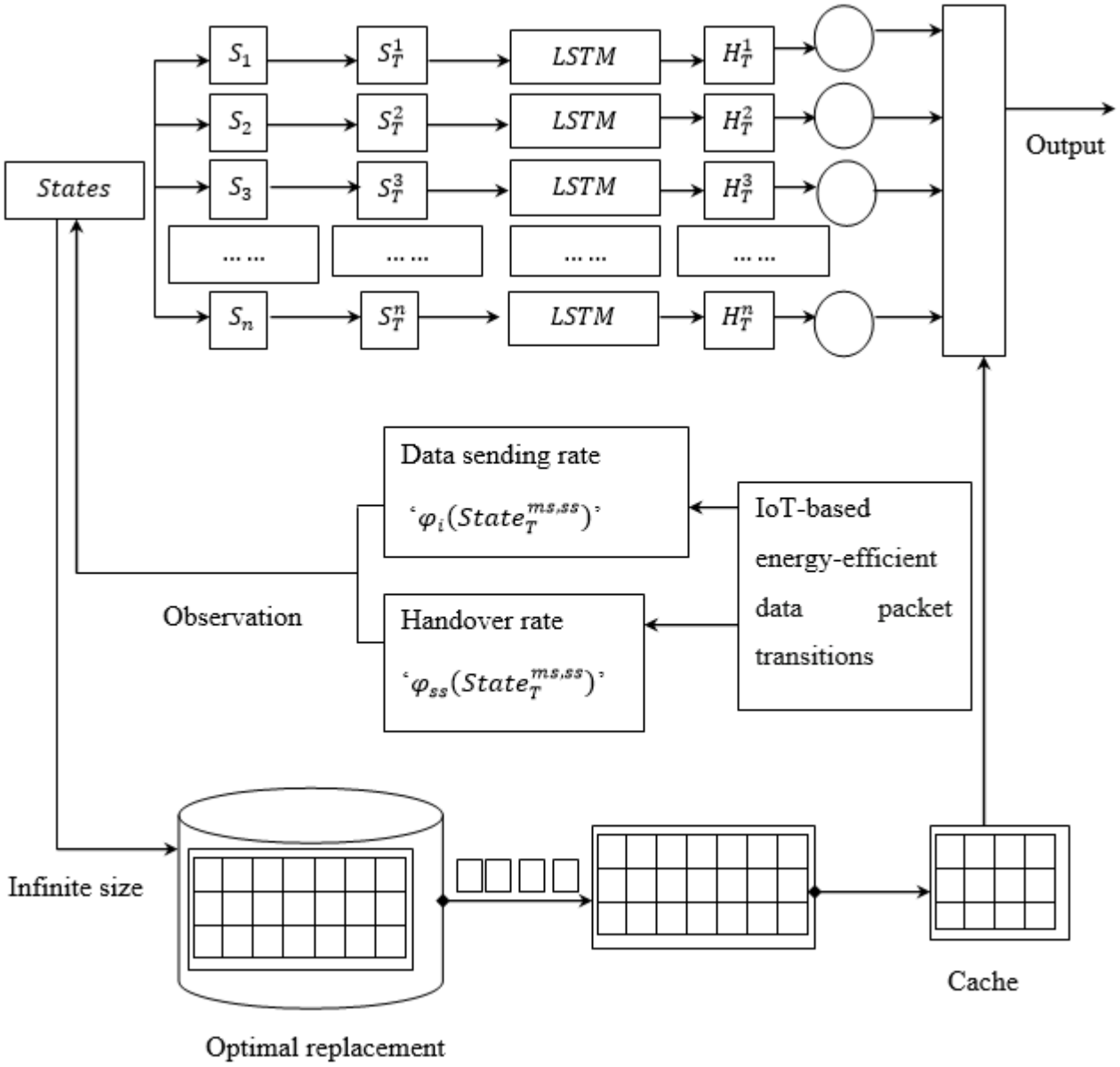

Figure 3 Block Diagram of Cache Obliviousness Neural Adaptive Congestion Control Data Transmission Model

As shown in the above figure 3, in the Cache Obliviousness Neural Adaptive Congestion Control Data Transmission model, the overall processes are split into four layers, namely, one hidden layer, one hidden layer, one stochastic layer, and finally one output layer. Here, with the input states obtained and proceeding to the input vector layer, the hidden layer performs the actual rule formation for each state. Next, in the stochastic layer, the cache obliviousness mechanism is applied to minimize the delay involved in controlling the congestion. Finally, the resultant output is provided in the output layer by means of output vector, therefore, ensuring either data transmission in case of a congestion-free network or else discarding the data transmission process in case of congestion. With this high goodput minimum time is said to be ensured.

At the time 'T', the proposed OUT-CONA method state for the corresponding nodes 'N' to be ready for sending data packets 'DP' with congestion control data transmission of a

RESEARCH ARTICLE

mainstream 'ms' and sub-stream represented by 'ss' without disconnecting the session is '$State_T^{ms} = \{State_T^{1,ss}, State_T^{2,ss}, \dots, State_T^{n,ss}\}$' and '$State_T = \{CWS_T^{ms,ss}, GP_T^{ms,ss}, SR_T^{ms,ss}\}$' respectively.

In addition, each '$State_T^{ms}$' is assigned with '$CWS_T^{ms,ss}$', '$GP_T^{ms,ss}$' and '$SR_T^{ms,ss}$' that represents the initialized congestion window size, goodput rate for congestion control data transmission, and sending rate of a mainstream 'ms' and sub-stream 'ss' respectively. Following this in the first layer (i.e., layer 1), an input state variable '$State_T^{ms,ss}$' is represented by each of the neurons within this layer. Finally, the state space of the entire network is progressed straightly to the input vector of the second layer. Second, layer 2 here represents the rule or the hidden layer. Each node of the second layer denotes the front end of an Ornstein–Uhlenbeck state transition scheduling model. The Normal Gaussian distribution of the nodes of this hidden layer is represented as given below.

$$HL_{ms,i}(State_T^{ms.ss}) = \exp\left[-\frac{\{(Rg_{ms} - Avg_{ms,i}(State_T^{ms,ss}))\}^2}{2\,W_{ms,i}^2}\right] \quad (6)$$

From the above equation (6), 'Rg_{ms}' and '$Avg_{ms,i}(State_T)$' represents the range of the entire network and the average value of Normal Gaussian distribution of the nodes within the mainstream 'ms' and sub-stream 'ss' for data transmission respectively. Then, for the state '$State_T^{ms.ss}$', for the mainstream 'ms' hidden layer, the Ornstein–Uhlenbeck Rule Stochastic is the Gaussian function product, represented as given below.

$$\varphi_i(State_T^{ms,ss}) = HL_{ms,i}\left(State_{ms,T}\right) = \exp\left[-\frac{\{(Rg_{ms,T} - Average_{ms,i}(State_T^{ms,ss}))\}^2}{2\,W_{ms,i}^2}\right] \quad (7)$$

From the above equation (7), '$\varphi_i(State_T^{ms,ss})$' represent the rule to be adopted during the handover between the operative and broadcasting model. Then, by means of Ornstein–Uhlenbeck rule fitness, optimal handover between operative and broadcasting mode is said to be ensured, therefore reducing the misclassification rate. The third layer is also referred to as the Normalized Stochastic Layer. The prime significance of this third Normalized Stochastic Layer is the consistent estimation of the fitness of each rule and normalization of the same. Then, the Normalized Stochastic function for '$i-th$' node is given below.

$$\psi_i(State_T^{ms,ss}) = \frac{\varphi_i(State_T^{ms,ss})}{\sum_{ss=1}^{n} \varphi_{ss}(State_T^{ms,ss})} \quad (8)$$

From the above equation (8), the Normalized Stochastic Layer results '$\psi_i(State_T^{ms,ss})$' are arrived at based on the handover between operative and broadcasting model '$\varphi_i(State_T^{ms,ss})$' with respect to '$i-th$' mainstream and sub-stream node data (i.e., data packet) sending rate and the corresponding handover between operative and broadcasting model '$\varphi_{ss}(State_T^{ms,ss})$' for '$n$' nodes respectively.

Finally, the fourth layer refers to the output layer. The OUT-CONA method output is made up of processor cache without possessing a predefined cache size for each block (i.e., each block representing a node's specified data packet at a time instance for transmission in the wireless network). Here, the node's data packet for analysis is initially split into smaller and smaller sub-streams in the form of blocks '$Block(DP_1) = DP_{11}, DP_{12}, \dots, DP_{1n}$'. The value function '$V(State_T^{ms,ss})$' for the corresponding blocks denotes the output of the block network, which is shown as:

$$V(State_T^{ms,ss}) \rightarrow Block(DP_i) = \sum_{i=1}^{n} W_{ij}\psi_i(State_T^{ms,ss}) \quad (9)$$

From the above equation (9), 'W_{ij}' denotes the weight between the cache output '$j-th$'and the hidden layer '$i-th$' node. In this manner, the above step is repeated by splitting the matrices in half and proceeding until the transpose of a matrix fits into the cache. In this manner, coherent shielding is said to be modeled between the organized and disorganized flow of data packets. Finally, the parameters obtained '$State_T$', 'W_i' and 'W_{ij}' are utilized in updating the range 'Rg' and average 'Avg' of the '$i-th$' hidden layer node in the OUT-CONA method is mathematically formulated as given below.

$$Rg_i(t+1) = Rg_i(t) + \frac{\alpha_{Rg_i}(1-\varphi_i)\,W_{ji}(State_T - Rg_i(t))}{\sigma_i^2} \quad 10)$$

$$Avg_i(t+1) = Avg_i(t) + \frac{\alpha_{Avg_i}(1-\varphi_i)\,W_{ji}(State_T - Rg_i(t))}{\sigma_i^2} \quad (11)$$

From the above equations (10) and (11), 'α_{Rg_i}', 'α_{Avg_i}' forms the learning rates for updating the range of the entire network and the average of nodes within mainstream 'ms' and sub-stream 'ss' for data transmission. As a result, the output of the block network greater than or equal to one is said to be free with congestion nodes, therefore ensuring smooth transmission between nodes. On contrary, the output of the block network with less than or equal to one is said to be nodes with congested nodes, therefore stopping transmission between nodes due to congestion. The pseudo-code representation of Cache Obliviousness Neural Adaptive Congestion Control Data Transmission is given in Algorithm 2.

Input: Dataset 'DS', nodes '$N = \{N_1, N_2, \dots, N_n\}$', data packets '$DP = \{DP_1, DP_2, \dots, DP_n\}$'

Output: congestion control-oriented robust goodput

RESEARCH ARTICLE

Step 1: Initialize time 'T', learning rate '$\alpha_{Rg_i} = 0.02, \alpha_{Avg_i} = 90$'

Step 2: Begin

Step 3: For each Dataset 'DS', data packets 'DP' with nodes 'N' and energy-efficient data packet transitions

//Layer 1 – input layer

Step 4: For each mainstream 'ms' and sub-stream 'ss'

Step 5: Formulate input state variable '$State_T^{ms,ss}$,

Step 6: End for

//Layer 2 – hidden layer

Step 7: For each mainstream 'ms' and sub-stream 'ss' of the nodes 'N' and data packets 'DP'

Step 8: Formulate Normal Gaussian distribution of the nodes as in equation (6)

Step 9: Evaluate Ornstein–Uhlenbeck rule fitness as in equation (7)

Step 10: End for

//Layer 3 – Normalized Stochastic Layer

Step 11: For each mainstream 'ms' and sub-stream 'ss' of nodes 'N' and data packets 'DP' with stochastic handover results to handle handover between operative and broadcasting model

Step 12: Evaluate Normalized Stochastic function for 'n nodes as in equation (8)

Step 13: End for

//Layer 4 – congestion control by means of cache obliviousness

Step 14: For each mainstream 'ms', sub-stream 'ss' of nodes 'N' and data packets 'DP' with normalized stochastic handover

Step 15: Obtain the output layer result as in equation (9)

Step 16: If '$V(State_T^{ms,ss}) \geq 1$'

Step 17: No congestion

Step 18: Data packet transmission between nodes or devices

Step 19: Update updating the range 'Rg' and average 'Avg' of the '$i-th$' hidden layer n as in equations (10) and (11)

Step 20: Proceed with other nodes data packets

Step 21: End if

Step 22: If '$V(State_T^{ms,ss}) < 1$'

Step 23: Congestion is observed

Step 24: No data packet transmission between nodes or devices

Step 25: Proceed with other nodes

Step 26: End if

Step 27: End for

Step 28: End for

Step 29: End

Algorithm 2 Cache Obliviousness Neural Adaptive Congestion Control Data Transmission

As given in the above Cache Obliviousness Neural Adaptive Congestion Control Data Transmission algorithm with the objective of improving the goodput rate with minimum delay, a LSTM-based neural adaptive with Cache Obliviousness mechanism is designed. The LSTM-based neural adaptive with the aid of four layers performs smooth data transmission via the adaptive nature, therefore improving the goodput rate. Next, by employing the Cache Obliviousness mechanism caching is reduced. This is owing to the reason that the size of the cache is not known in advance, the blocks of data packets to be transmitted are split in a recursive manner with further subdivisions in the cache. By employing this split approach, a similar level of complexity is said to be arrived at for the overall matrix. With this goodput is said to be attained with minimum end-to-end delay.

4. EXPERIMENTAL SETUP

This section discusses the in-depth analysis of the proposed Ornstein–Uhlenbeck Transition and Cache Obliviousness Neural Adaptive (OUT-CONA) congestion control for Internet of Things and existing methods, adaptive aggregation [1], and hybrid deep learning [2] using four constraints, energy consumption, misclassification rate, goodput, and end-to-end delay respectively. The proposed OUT-CONA method and the existing work were evaluated using Python tool by employing IoT Traffic Generation Patterns Dataset (https://www.kaggle.com/datasets/tubitak1001118e277/iot-traffic-generation-patterns). The data from the IoT Traffic Generation Patterns Dataset have been obtained in the laboratory at Yasar University as part of The Scientific and Technological Research Council of Turkey (TUBITAK). Here, the first raw sensor readings were collected at the laboratory from sensors. Second, the data set for each such sensor has been transformed into bits with the purpose of forming the traffic generation pattern of each IoT device. Third and finally, the data set for 10,000 IoT devices have been modeled from the traffic generation patterns of these respective IoT devices. The IoT Traffic Generation Patterns Dataset consists of a structure array in the MATLAB structure format comprising 10001 rows and 5 fields. Here, the rows represent the IoT devices, where the first row is left empty

RESEARCH ARTICLE

and in addition to each row, the following fields are provided in the table 2.

Table 2 IoT Traffic Generation Patterns Dataset Description

S. No	Feature	Description
1	Data (KB)	The data here represents the time-series data that represents the traffic generation pattern of respective device, measured in terms of kilobits.
2	Data rate (kbps)	The data rate represents the corresponding IoT device rate of data whose unit is measured in terms of kilobits per second.
3	Start time (seconds)	The start time offset refers to the corresponding IoT device starts generation.
4	Sampling period (seconds)	The sampling period denotes the inter arrival time between traffic generation of corresponding IoT device.
5	Delay constraint (seconds)	The delay constraint represents the IoT device that should send its traffic.

4.1. Performance Analysis of Energy Consumption

Energy consumption represents the amount of energy consumed for analyzing corresponding network traffic congestion toward robust data transmission. To be more specific, the energy consumption rate refers to the total amount of energy consumed towards ensuring congestion control data transmission between nodes or devices. Energy conservation in turn, therefore, results in more efficient mechanisms hence reducing the congestion in wireless networks.

$$EC = \sum_{i=1}^{n} DP_i * EC\left(\frac{dDP_t}{dt}\right) \quad (12)$$

From the above equation (12), energy consumption 'EC' is measured by means of data packets being ready for transmission in a wireless network and based on the data packets being sent by the devices or nodes 'DP_i' and the actual energy consumed for congestion control data transmission '$EC\left(\frac{dDP_t}{dt}\right)$'. It is measured in terms of joules 'J'. Table 3 explains the energy consumption of various techniques.

Table 3 Energy consumption using OUT-CONA, Adaptive aggregation [1] and Hybrid deep learning [2]

Data Packets	Energy Consumption (J)		
	OUT-CONA	Adaptive Aggregation	Hybrid Deep Learning
25	0.875	1.075	1.225
50	1.025	1.435	1.815
75	1.345	1.825	2.135
100	1.525	2.015	2.245
125	1.845	2.234	2.855
150	2.315	2.535	3.215
175	2.455	2.815	3.535
200	2.721	3.215	3.825
225	2.945	3.455	4.155
250	3.355	3.815	4.525

Figure 4 illustrates energy consumption using 250 various data packets considered in the x-axis, calculated in joules. From the above figure, the number of data packets is found to be directly compared to energy consumption. While enhancing the data packets, state transitions between nodes or devices for transmission enhance as well therefore, energy consumption is enhanced. Consumption of energy is defined as the actual energy consumed for controlling traffic or congestion among nodes.

An OUT-CONA technique illustrates minimal energy consumed compared with congestion control, Adaptive aggregation [1] as well as Hybrid deep learning respectively. To be more specific, the proposed OUT-CONA minimizes the energy consumption of data packets is lesser. Experimental evaluation of 25 data packets, energy consumption with OUT-CONA is '0.875J', '1.075J', and '1.225J' with [1] [2]. From the graph, the higher achievement using OUT-CONA of Ornstein–Uhlenbeck state energy-efficient transition significantly classifies radio node status (i.e., operative, idle, or broadcasting) based on various operating modes. Gauss and Markov's processes are utilized in obtaining the probabilistic node state transitions based on the probabilistic node state transitions via the Gauss process and multiple approximations made by means of Markov process, conditional distributions are attained. From this, the consumption of energy is said to be minimized by 18% and 32% compared to [1] [2]. From the above results, it is inferred that the OUT-CONA SC method outcomes within a lesser packet drop rate for minimizing congestion or controlling the congestion significantly.

International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2023/218512 Volume 10, Issue 1, January – February (2023)

RESEARCH ARTICLE

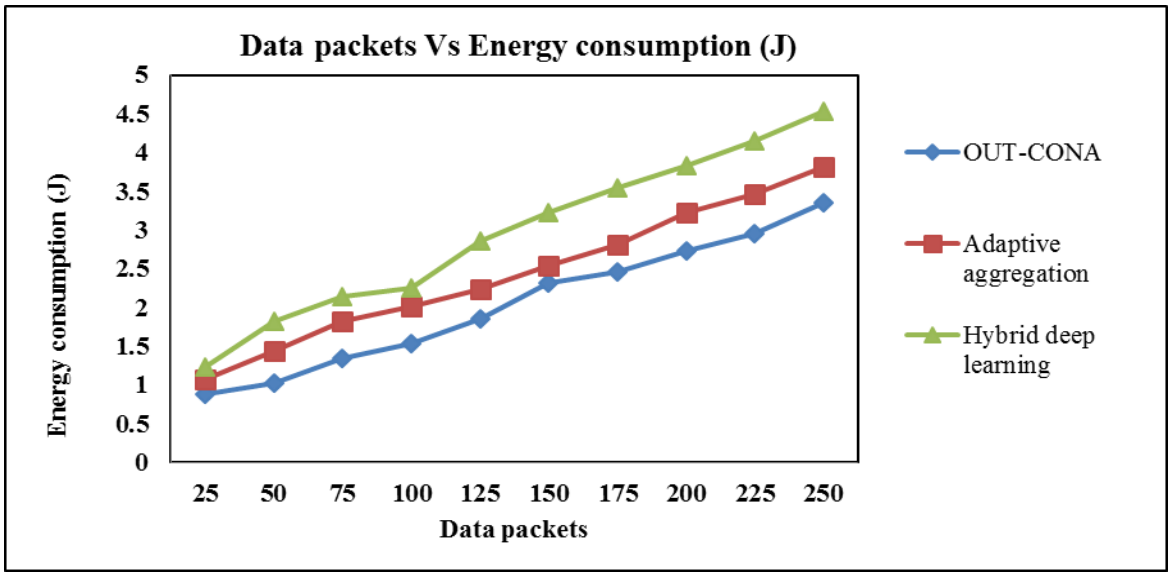

Figure 4 Analysis of Energy Consumption with Increasing Data Packet

4.2. Performance Analysis of Misclassification Rate

During handover i.e., between active and broadcasting modes, a certain amount of misclassification is said to take place (i.e., active mode is considered to be broadcasting mode and vice versa), therefore causing congestion.

$$MR = \sum_{i=1}^{n} \frac{N_{O \leftrightarrow B}}{N_i} * 100 \qquad (13)$$

In (13), the misclassification rate 'MR' is calculated on nodes or the devices involved in the simulation of congestion control data transmission in wireless network 'N_i' and the nodes in the handover stage '$N_{O \leftrightarrow B}$' between operative 'O' and broadcasting 'B' mode respectively. It is measured in terms of percentage. Table 4 illustrates the misclassification rate with various techniques.

Table 4 Misclassification Rate Using OUT-CONA, Adaptive Aggregation [1] and Hybrid Deep Learning [2]

Nodes	Misclassification rate (%)		
	OUT-CONA	Adaptive Aggregation	Hybrid Deep Learning
50	88	92	94
100	82.15	86.65	93.25
150	82	86	92
200	81.35	83.25	91.85
250	78.55	86	91.35
300	75.35	81.55	91
350	73.25	80.35	90.45
400	71.45	79	90
450	71	79	88.35
500	70.25	77	87.55

Figure 5 given above shows the misclassification rate with respect to the increasing nodes ranging between 50 and 500. With x-axis represents the number of nodes or devices involved in the communication process, the y-axis denotes the misclassification rate involved in the analysis of congestion control during data transmission. From the above figure, the misclassification using the OUT-CONA method was found to be improved upon comparison with [1] and [2]. This is inferred from the simulation results, with 50 nodes involved in the simulation process, '47' nodes performed the correct handover between operative 'O' and broadcasting 'B' mode using the OUT-CONA method, '46' nodes performed the correct handover between operative 'O' and broadcasting 'B' mode using [1] and '44' nodes performed correct handover between operative 'O' and broadcasting 'B' mode using [2]. This confirms the results. The reason behind the minimization of the misclassification rate using the OUT-CONA method was due to the application of the Ornstein–Uhlenbeck State Energy-efficient Transition algorithm. By applying this algorithm, first, operative and broadcasting modes were identified by means of Gauss and Markov processes separately. Next, the identified modes handover between operative and broadcasting were differentiated via the Percolation Centrality function. By applying this function, data sessions were only transferred between states without explicit disconnection, therefore reducing the misclassification rate using the OUT-CONA method by 7% and 15% compared to [1] [2].

International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2023/218512 Volume 10, Issue 1, January – February (2023)

RESEARCH ARTICLE

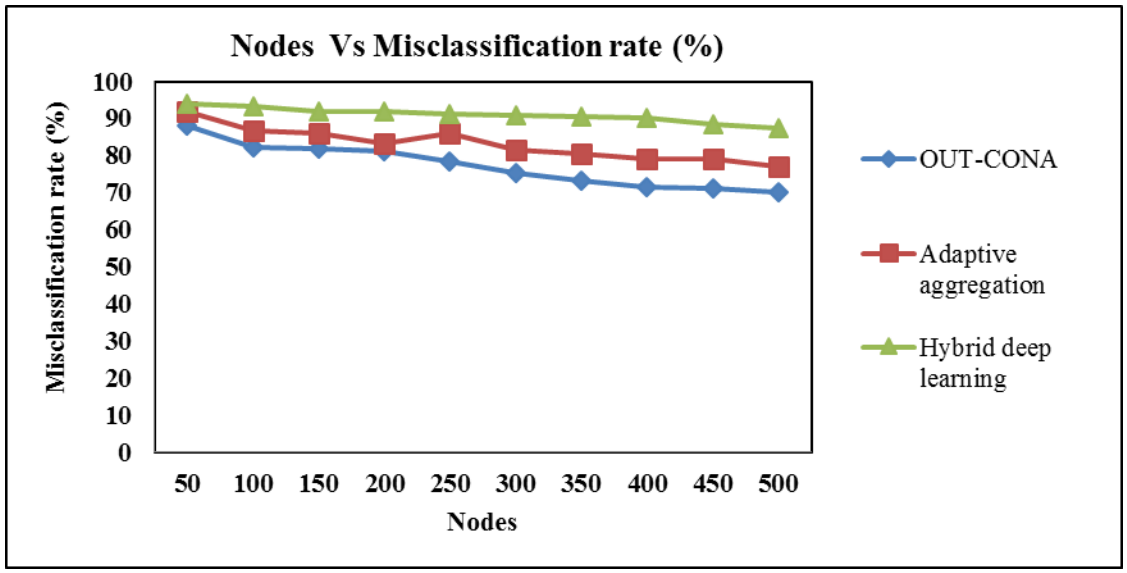

Figure 5 Analysis of Misclassification Rate with Increasing Nodes

4.3. Performance Analysis of Goodput

Goodput is similar to throughput in that it estimates the speed of data traveling the network from its starting point to its destination. Moreover, it estimates how fast and accurately data traverses the network and arrives at its desired location. It is evaluated by dividing the original or actual data divided by the transfer time.

$$GP = \frac{DP_{size}}{Transfer_T} \quad (14)$$

From the above equation (14), the rate of goodput 'GP' is measured on the basis of the size of data packet 'DP_{size}' and the transfer time '$Transfer_T$'. It is measured in terms of megabytes per second. For example, if a node consisting of 5-megabyte file may require 300 kilobytes of header information as well as acknowledgments to be transferred through the data transfer procedure, goodput is said to be the original 5 megabytes separated with the transfer time. Table 5 illustrates goodput values of various techniques.

Table 5 Goodput Rate Using OUT-CONA, Adaptive Aggregation [1] and Hybrid Deep Learning [2]

Nodes	Goodput (%)		
	OUT-CONA	Adaptive Aggregation	Hybrid Deep Learning
50	3.33	2.94	2.63
100	3.85	3.15	2.85
150	4.25	3.85	3.35
200	4.95	4.15	3.55
250	5.35	4.45	3.95
300	6.15	4.85	4.15
350	6.35	5.25	4.55
400	6.95	5.35	4.95
450	7.25	5.85	5.25
500	7.55	6	5.85

Figure 6 given above shows the goodput rate of OUT-CONA, Adaptive aggregation [1] as well as Hybrid deep learning [2] in accordance with the different numbers of nodes or devices. As shown in the above figure, increasing the number of nodes, higher numbers of nodes get collected within the cache as well as congestion. As a result, nodes are directly relative to the goodput rate. Increasing the number of nodes involved in the simulation process, considerably only a smaller amount of increase in goodput is said to occur. Experimental evaluation of 50 nodes, it was found that '1.5s' were involved in the data transfer using OUT-CONA, '1.7s' were involved in the data transfer using [1] and '1.9s' were involved in the data transfer using [2]. With this, the rate of goodput was observed to be '3.33Mbps', '2.94Mbps', and '2.63Mbps' using OUT-CONA, [1, 2] respectively. From the above results, it is inferred that the OUT-CONA attains better congestion control upon comparison with [1, 2]. The reason the OUT-CONA method combines factors is analyzed using the Cache Obliviousness-based Congestion mechanism. From utilizing an integrated algorithm, measures are focused as well hence ensuring a better rate of goodput. With this, the goodput rate using the OUT-CONA method was comparatively higher by 21% and 36% compared to [1] [2].

RESEARCH ARTICLE

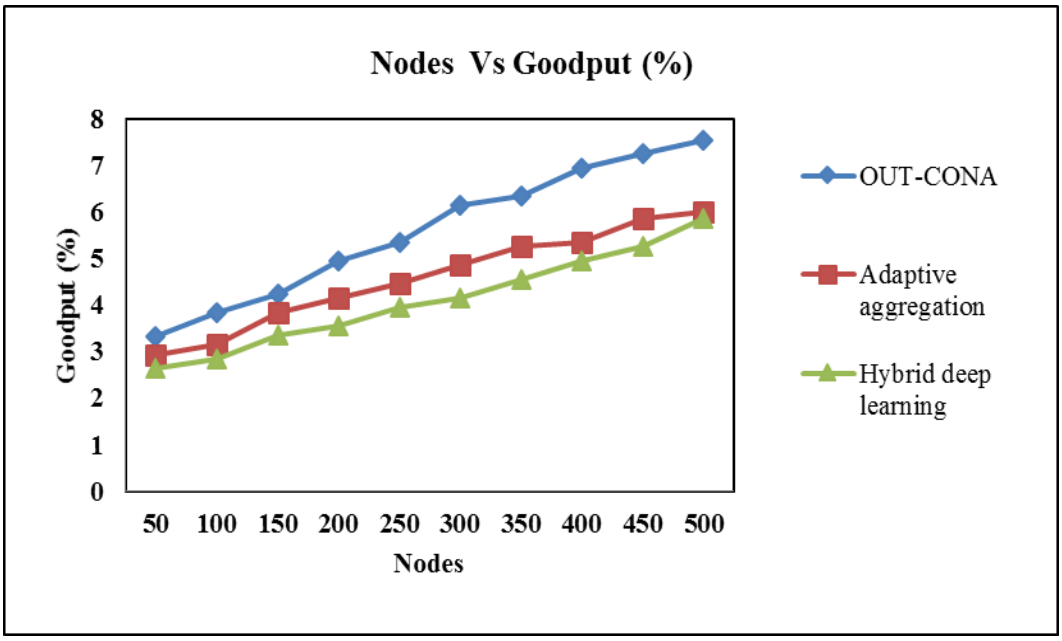

Figure 6 Analysis of Goodput Rate with Increasing Nodes

4.4. Performance Analysis of End-to-End Delay

It is defined as the time consumed to transfer the packet across the network (i.e. IoT-enabled wireless network) between the source and destination node. It is evaluated as given below.

$$E2ED = \sum_{i=1}^{n} DP_i * Time\ [S \rightarrow D] \quad (15)$$

In equation (15), '$E2ED$' is evaluated on the basis of the number of data packets involved in the simulation towards congestion control data transmission in wireless network 'DP_i' and time is taken for transmitting the data packets '$Time\ [S \rightarrow D]$' respectively. It is measured in terms of milliseconds (ms). Table 6 illustrates end-to-end delay values for various techniques.

Table 6 End-to-End Delay Using OUT-CONA, Adaptive Aggregation [1] and Hybrid Deep Learning [2]

Data Packets	End-to-End Delay (ms)		
	OUT-CONA	Adaptive Aggregation	Hybrid Deep Learning
25	8.75	11.25	12.5
50	10.45	13.35	17.55
75	11.25	21.35	25.35
100	13.55	25.25	31.35
125	18.25	28.15	35.35
150	19.35	30.35	40.55
175	21.45	35.45	43.15
200	23.25	38.25	48.25
225	25	41.35	55.25
250	28.55	45.55	60

Figure 7 shows the delay observed for controlling congestion with OUT-CONA and Adaptive aggregation [1] as well as Hybrid deep learning [2] respectively. In the figure, data packet increase causes the involvement of better IoT-enabled nodes in a wireless network. Simulation evaluation of 25 data packets of delay was attained as '8.75 ms' with OUT-CONA, '11.25 ms' with [1] as well as '12.5 ms' with [2].

Experimental evaluation is attained for the OUT-CONA method to achieve the higher performance of the two methods in average delay. The enhancement is obvious as data packets shoot up, where time consumption of data packets traveling among source as well as a destination goes through the ceiling. As specified before, the OUT-CONA method takes into consideration of each data packet transmission.

Though, Adaptive aggregation [1] schedules data packets on IoT traffic types, as the Hybrid deep learning [2] data packets on hybrid AI-based solution without considering the end-to-end path situation. Owing to this reason, existing methods are more highly sensitive than OUT-CONA with a number of hops. It degrades the delay performance by 26% and 49% compared to [1] [2].

RESEARCH ARTICLE

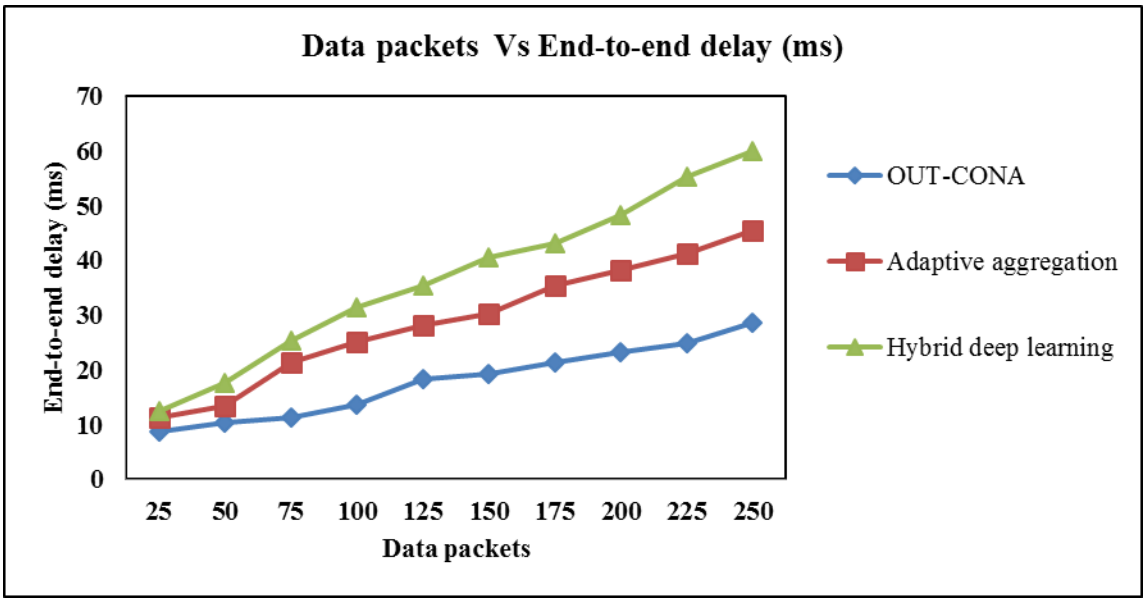

Figure 7 Analysis of End-to-End Delay with Increasing Data Packets

5. CONCLUSION

In this paper, an Ornstein–Uhlenbeck Transition and Cache Obliviousness Neural Adaptive (OUT-CONA) congestion control method for efficient data transmission in a wireless network is proposed. To start with, a novel Ornstein–Uhlenbeck State Transition scheduling model is designed to differentiate between three distinct modes. Next, the Gauss and Markov Processes based Estimation of State Transition along with the Percolation Centrality function is applied to both reduce the energy consumption and misclassification rate. Next, the Cache Obliviousness Neural Adaptive Congestion Control Data Transmission algorithm is designed to solve the congestion by addressing overflow and providing via data sending rate and handover rates for IoT-enabled wireless networks. By utilizing blocks of data packets to be transmitted are split in a recursive manner with further subdivisions in the cache, therefore, improving the rate of goodput with minimum end-to-end delay. Simulation results verify and validate the efficiency of the proposed method in terms of energy consumption, misclassification rate, goodput, and end-to-end delay compared to the existing methods.

REFERENCES

[1] Amin S. Ibrahim, Khaled Y. Youssef, Ahmed H. Eldeeb, Mohamed Abouelatta, Hesham Kamel, “Adaptive aggregation based IoT traffic patterns for optimizing smart city network performance”, Alexandria Engineering Journal, Elsevier, Volume 61, Issue 12, December 2022, Pages 9553-9568.

[2] Sulaiman Khan, Anwar Hussain, Shah Nazir, Fazlullah Khan, Ammar Oad, Mohammad Dahman Alshehr, “Efficient and reliable hybrid deep learning-enabled model for congestion control in 5G/6G networks”, Computer Communications, Elsevier, Volume 182, January 2022, Pages 31-40.

[3] Thilina N. Weerasinghe, Indika A. M. Balapuwaduge, Frank Y. Li, “Priority-based initial access for URLLC traffic in massive IoT networks: Schemes and performance analysis”, Computer Networks, Elsevier, Volume 178, September 2020, Pages 1-16.

[4] Muhammad Adi, “Congestion free opportunistic multipath routing load balancing scheme for Internet of Things (IoT)”, Computer Networks, Elsevier, Volume 184, January 2021, Pages 1-27.

[5] Fan Wang, Min Zhu, Maoli Wang , Mohammad R. Khosravi, Qiang Ni , Senior Member, IEEE, Shui Yu , Senior Member, IEEE, and Lianyong Qi, “6G-Enabled Short-Term Forecasting for Large-Scale Traffic Flow in Massive IoT Based on Time-Aware Locality-Sensitive Hashing”, IEEE Internet of Things Journal, Volume 8, Issue 7, April 2021, Pages 5321 - 5331.

[6] Lal Pratap Verma, Mahesh Kumar, “An IoT based Congestion Control Algorithm”, Internet of Things, Elsevier, Volume 9, March 2020, Pages 1-24.

[7] M. Swarna, T. Godhavari, “Enhancement of CoAP based congestion control in IoT network - a novel approach”, Materials Today: Proceedings, Elsevier, Volume 37, Issue 2, June 2020, Pages 775-784.

[8] Mohammed Aljubayri Zhaohui Yang Mohammad Shikh-Bahaei, “Cross-layer multipath congestion control, routing and scheduling design in ad hoc wireless networks”, Wiley, Volume 2021, Feb 2021, Pages 1-13.

[9] Youngjae Park and Sungwook Kim, “Game-based data offloading scheme for IoT system traffic congestion problems”, EURASIP Journal on Wireless Communications and Networking, Springer, Volume 198, October 2015, Pages 1-10.

[10] Shilpa P. Khedkar, R. Aroul Canessane, Moslem Lari Najafi, “Prediction of Traffic Generated by IoT Devices Using Statistical Learning Time Series Algorithms”, Wireless Communications and Mobile Computing, Wiley, Volume 2021, August 2021, Pages 1-12.

[11] Dawei Shen, Wei Yan, Yuhuai Peng, Yanhua Fu, and Qingxu Deng, “Congestion Control and Traffic Scheduling for Collaborative Crowdsourcing in SDN Enabled Mobile Wireless Networks”, Wireless Communications and Mobile Computing, Wiley, Volume 2018, February 2018, Pages 1-11.

[12] Muhannad Quwaider, Yousef Shatnawi, “Congestion control model for securing internet of things data flow”, Ad Hoc Networks, Elsevier, Volume 108, May 2020, Pages 1-12.

[13] Amin S. Ibrahim, Khaled Y Youssef, Mohamed Abouelatta, “Traffic Aggregation Techniques for Optimizing IoT Networks”, Advances in

RESEARCH ARTICLE

Science, Technology and Engineering Systems Journal, Volume 6, Issue 1, June 2021, Pages 509-518.
[14] Juan Pablo Astudillo Leon, Francisco J. Rico-Novella, Luis J. De La Cruz Lllopis, "Predictive Traffic Control and Differentiation on Smart Grid Neighborhood Area Networks", IEEE Access, Volume 6, Issue 1, December 2020, Pages 216805 – 216821.
[15] Farhad Hassan, Amir Ijaz, Mubashir Ali, Zeshan Afzal, FarrukhArslan, "Iot Enabled Intelligent Traffic Congestion Handling System Empowered By Machine Learning", International Journal of Scientific & Technology Research, Volume 10, Issue 06, June 2021, Pages 1-5.
[16] Sadaf Mokhtari, Hamid Barati, Alli Barati, "A Hierarchical Congestion Control Method in Clustered Internet of Things", The Journal of Supercomputing, Springer, Volume 78, February 2022, Pages 11830–11855.
[17] Godfrey A. Akpakwu1 Gerhard P. Hancke, Adnan M. Abu-Mahfouz, "CACC: Context-aware congestion control approach for lightweight CoAP/UDP-based Internet of Things traffic", Transactions of emerging telecommunications technologies, Volume 31, Issue 2, October 2019.
[18] Phet Aimtongkham, Tri Gia Nguyen, and Chakchai So-In, "Congestion Control and Prediction Schemes Using Fuzzy Logic System with Adaptive Membership Function in Wireless Sensor Networks", Wireless Communications and Mobile Computing, Wiley, Volume 2018, August 2018, Pages 1-12.
[19] Zhi Hu, Xiaowei Wang, Yuxia Bie, "Game Theory Based Congestion Control for Routing in Wireless Sensor Networks", IEEE Access, Volume 9, July 2021, Pages 103862 – 103874.
[20] Chansook Lim, "Improving Congestion Control of TCP for Constrained IoT Networks", Sensors, Volume 20, Issue 17, August 2020, Pages 1-16.

Authors

Dr. N Thrimoorthy is currently working as Assistant Professor in the school of CSA, REVA University, Bengaluru, India. He has done his Bachelors and Post-Graduation from Dravidian University in the years 2003 and 2005 respectively from Dravidian University. He also completed his M.Tech from Acharya Nagarjuna University in the year 2010 and Ph.D from Dravidian University in year 2019. He is research interests includes Wireless Sensor Networks and Internet of Things. Email: thrimoorthy.n@reva.edu.in.

Dr. Somashekhara Reddy D is working as a assistant professor in department of computer science and engineering in Jain University, Global campus, Bangalore. He has completed his PhD from Periyar University Salem, Tamil nadu. His research area is wireless and Mobile communication. Email: somashekhara.mtech@gmail.com

Dr. Chandramma R, Assistant Professor in Department of computer science and engineering in Jain University, Global campus, Bangalore. She has completed her PhD from VTU, Karnataka. Her research area is wireless and Mobile communication and Machine Learning Email: chandramma.cse@gmail.com

Ms. Soumya Unnikrishnan is working as an Assistant Professor in the Department of Computer Science and Applications, St. Francis College, Koramangala, Bangalore. Currently she is pursuing her PhD at Christ (Deemed to be University), Bangalore. Her research interests are Artificial Intelligence and Networking.
Email: soumya.u@stfranciscollege.edu.in

Dr. K. Vanitha, Assistant Professor in Department of Computer Science and Engineering in Jain University, Global campus, Bangalore. She has completed her PhD from Anna University Chennai, TN. Her research area is wireless and Mobile communication and Networks Security.
Email: profvanithacse@gmail.com

How to cite this article:

N Thrimoorthy, Somashekhara Reddy D, Chandramma R, Soumya Unnikrishnan, Vanitha K, "Ornstein Uhlenbeck Cache Obliviousness Neural Congestion Control in Wireless Network for IOT Data Transmission", International Journal of Computer Networks and Applications (IJCNA), 10(1), PP: 68-83, 2023, DOI: 10.22247/ijcna/2023/218512.