Markov transition and smart cache congestion control for IoT enabled wireless mesh networks

Yuvaraj N & Saravanan G

Peer-to-Peer Networking and Applications

ISSN 1936-6442

Peer-to-Peer Netw. Appl.
DOI 10.1007/s12083-020-00969-4

ONLINE FIRST

Springer

Peer-to-Peer Networking and Applications
https://doi.org/10.1007/s12083-020-00969-4

Markov transition and smart cache congestion control for IoT enabled wireless mesh networks

Yuvaraj N[1] · Saravanan G[2]

Received: 24 March 2020 / Accepted: 21 July 2020

Abstract
Wireless Mesh Networking (WMN) is the latest Internet framework that provides a comprehensive range for Subsequent Internet (SI) prototype. Despite significant advantages provided by WMN, its practical distribution to connect Internet of Things (IoT) networks caused exorbitant congestion and restricted bandwidth. Motivated by this, a novel mechanism that ensures control in the manifestation of mobbing, end-to-end delay, energy consumption for enhancing the network performance of IoT-enabled WMN is presented. The proposed method is called as an Integrated Markov State Transition and Open Loop Smart Caching (MST-OLSC) for congestion control in IoT-enabled WMN. The proposed method uses Markov state transition scheduling model to differentiate the states of the incoming data packets from the host computer. This is performed by applying the State Betweenness centrality. Next, Congestion Control Token Caching mechanism is applied with the objective of controlling the congestion by means of caching via overflow with well-balanced isolation between regulated and unregulated flow of data packet. Finally, Open Loop Smart Caching is presented to ensure constant data rate, thereby providing fair inflow and outflow between the incoming and outgoing data packets. The evaluation results of MST-OLSC ensure higher network performance with minimum end-to-end delay, energy consumption and higher packet delivery rate is achieved with respect to inflated IoT nodes in WMN.

Keywords Wireless mesh networking · Internet of things · Markov state transition · Congestion control · Smart caching · State Betweenness centrality

1 Introduction

Over the coming few years, the data flow in internet was anticipated to enlarge exponentially with the increase in the frequency of coupled portable computing devices. Due to this, the frequency of coupled portable computing devices and communications between devices are forecasted to outshine the overall population over the coming years. Hence, there is requirement for an advanced Internet infrastructure to discourse the Quality of Service (QoS) requirements in maintaining the increase in Internet traffic. A Machine learning technique with Hierarchical Recurrent Random Neural Network (Hierarchical RNN) was presented in [1]. The technique was split into two levels, called, top-level RNN and bottom-level RNN. Optimal routes were selected by means of bottom-level RNNs and best routes were selected by the top-level RNN. According to the routing based on top-level RNN or bottom-level RNN, different types of cognitive data packets were routed. With this only the bottom RNNs were trained individually and the quality of service in terms of routing and throughput were said to be ensured. Despite these advantages, the packet delivery ratio remains lower.

Cognitive Heterogeneous Routing (CHR) was proposed in [2] enabled heterogeneous types of network to function as a single network. Here, the data packets were routed based on QoS metrics. Finally, transmission optimization was performed with the aid of a novel learning pattern using reinforced routing via Q-learning. This was applied with the

✉ Yuvaraj N
yuvait09@gmail.com

Saravanan G
gsaravanan.pacet@gmail.com

[1] Department of Information Technology, J.K.K.Nattraja College of Engineering and Technology, NH-544 Natarajapuram, Komarapalayam, Tamil Nadu 638183, India

[2] Department of Computer Science and Engineering, Erode Sengunthar Engineering College, Erode, Tamil Nadu 638057, India

Published online: 12 August 2020

objective of improving network performance. Besides, the heterogeneous type of network also used Internet of Things for providing the objective of smart homes and smart cities.

In this work, a mechanism involving management of congestion for WMN in the context of IoT is designed. Specifically in WMN sensor enabled network scenario in the context of IoT, the proposed WMN method utilizes radio sensors for sending the data packets. To minimize the delay incurred for each sensor, the method also measures the status of each and every sensor's using a Variable Markov-based scheduling and performs the actions in an appropriate manner, i.e. data packets transmission between the nodes in WMN. Though transmission optimization was achieved, delay incurred was found to be higher. The proposed work contributions are listed as below:

- The Markov State Transition theory is employed for classification of sensor node status in the context of IoT into three distinct controlling procedures: active '*A*', inactive '*IA*' and broadcast '*B*' for the flexible sensor scheduling model. This distinct controlling procedure identifies node possessing bandwidth with minimum end-to-end delay, specifically in the case of inactive mode. The sensor scheduling model was applied over a proposed WMN IoT where only generation of data packets between nodes to be sent is activated.
- We designed an Open Loop Smart Caching congestion control mechanism to save the maximum size of the bucket so that overflow of data packet is avoided by allowing only the data packets to be processed via buffer according to the availability of token. By following this pattern, the congestion incurred in IoT-enabled WMN is greatly reduced with optimal data packets accepted in the buffer.
- In data packet routing scheme, we used an integrated Markov State Transition and Open Loop Caching to address both the overflow and data constant rate. State Betweenness Centrality is employed to apply different states for different transition pattern according to the conditional distribution. This in turn provided more control over the network and energy consumption is reduced. In addition to the application of the caching mechanism, this data packet routing scheme aims to reduce the energy consumption in IoT-enabled WMN in a significant manner. The flow control mechanism aims to minimize the occurrence of delay via constant data flow rate in IoT-enabled WMN in an extensive manner.

The proposed method is organized as follows: section 2 reviews the work concerned in IoT involving congestion control mechanism, the proposed method with the network model, followed by system model to present a congestion control mechanism is elaborated in Section 3. Simulation setup to carry out experiments and an elaborate discussion with the evaluation results is provided in Section 4. Finally, concluding remarks are provided in Section 6.

2 Review of related literature

In the current scenario, the utilization of a single Access Point (AP) to obtain the information pertaining to Internet is performed through inclusive and non-inclusive utilization of real-time requirements. In these circumstances the AP bandwidth is said to be allocated to the client based on the propositions that are not said to be persuaded with the familiar concerns associated with priority factors of flow of internet.

An innovative method for controlling the traffic with the objective of improving the throughput and minimizing the delay was presented in [3]. However, the paradigm shift in IoT has resulted in significant increment of communication and computation capabilities. To address this issue, Edge Mesh computing paradigm was designed in [4] to describe task management and task allocation. However due to scaling improvement in device frequency connected in the network increased the latency rate. An Access Pattern Analyzer (APA) was designed in [5] that measured the features of traffic in an intermittent manner, therefore reducing the random access load.

However, these basic IoT functions specifically targeted stable environments as a powerful requirement for next generation of IoT services to be quicker and more authentic. In [6], Elastic IoT was presented using Artificial Intelligence technologies. With this, packet delivery rate along with the round trip time was said to be reduced. With larger number of heterogeneous devices addressed in IoT and due to the presence of un-trusted networks, security is said to be compromised. To address this issue, a game theoretic technique was introduced in [7] that detected intruders via signature detection technique. With this, a trade-off between detection and false positive rate was said to be attained.

Future generation mechanisms involving accessing of network and various services available via Internet have enlarged the provocations of supposing quality of service for nodes with conventional control models for congestion. A framework based on the design of machine learning technique named as Reinforcement Q-Learning was designed in [8] which utilizes Kanerva Coding function approximation algorithm to reduce the computational complexity and transmission latency. However, energy consumption was not resolved. To address this issue, a provably correct scheduling reliable data delivery was presented in [9], which achieved perpetual communications besides addressing energy consumption issues.

Yet another emergency energy aware congestion control model was designed in [10] utilizing multi-channel allocation technique. With this emergency aware congestion control mechanism, an extensive advancement was said to be achieved in terms of packet delivery, throughput concerning network and passage time. In multi-radio multi-channel WMN, the capacity of pertaining each network necessitates as a complicated design in cross-layer pattern.

A price-based framework was designed in [11] to address factors associated with control of congestion in a distributed pattern and assigning the channels via localized pattern. With this different fairness objectives were achieved. Yet another advanced congestion control algorithm was proposed in [12] using greedy mechanism. In this algorithm, the number of retransmissions was reduced by guaranteeing throughput and delay. However with complicated traffic, congestion rate remains higher. To address this issue, a Load Shifting Technique was presented in [13] that in turn reassigned load with the purpose of reducing the congestion rate.

One of the challenging tasks in WMN is concerning delivery of the corresponding data owing to the link quality changes, higher rate of interference and congestion occurrences. In [14], a method concerning optimization was designed for Dynamic Traffic Engineering (O-DTE) to reduce the interference and rate of congestion to attain excessive data delivery. Yet another work to address interference was presented in [15] using nested optimization strategy. Along with this, a genetic approach was designed for fair allocation of channels, route scheduling and optimal path selection. With this, the scheduling efficiency was improved. However, the load aspect was not considered. In [16], load factor based on congestion control mechanism was designed via window adjustment policies.

A joint measurement of the overall performance of the service and control mechanism for congestion was designed in [17] by means of adaptive ACK scheme to enhance the network significance in terms of packet dispatch, delay incurred in transmission and good-put. An explicit flow control mechanism was presented in [18] using congestion control in real time. A Light Weight Datagram Congestion Control Protocol (LW-DCCP) was designed in [19] by means of caching and retransmission algorithms to attain better performances in terms of delay incurred during transmission, amount of energy consumed and peak signal to noise ratio.

Motivated by the above factors, we pin at the same time and evaluate our proposals towards congestion control for IoT-enabled WMN, considering multi-constraints of routing such as, linkage stability, end-to-end delay and bandwidth. The congestion control mechanism takes into account three different states, active, inactive and broadcast. Then, classifications are made accordingly. Each of these states is activated via Markov State Transition model. Next, according to the availability of tokens in the buffer, the data packet is accepted from the host computer, therefore addressing overflow via Open Loop Smart Caching.

3 Proposed method

The elaborate description of Markov State Transition and Open Loop Smart Caching (MST-OLSC) for congestion control in IoT-enabled WMN is provided below.

3.1 System model overview

Congestion control is said to be a state happening in network layer as message traffic is so massive which decelerates overall network response time. With the increase in delay, the performance is said to be reduced, resulting in retransmission, causing circumstances unpleasant. In this section, a network and system model used to design a congestion control IoT-enabled Wireless Mesh Network, using the method, Markov State Transition and Open Loop Smart Caching (MST-OLSC).

3.2 Network model

Wireless Mesh Network comprises of radio nodes '$N = \{N_1, N_2, \ldots, N_n\}$' grouped in the form of mesh topology that distribute data packets '$DP = \{DP_1, DP_2, \ldots, DP_n\}$' between each other. The connection and distribution of data packets between radio nodes were established via routing table. The content of the routing table is shown in Fig. 1.

As given in the above routing table, the vital information to be provided are the origin qualifier 'OQ', the target qualifier 'TQ', origin series number 'OSN', target series number 'TSN', telecast qualifier 'TQ' and time to hangout 'TTH' of the corresponding nodes. Using the above contents in the routing table, a congestion control mechanism based on the system model given below is structured.

3.3 System model

The design of the MST-OLSC method is based on the diversified-control path algorithm. The network of mesh routers is modeled by means of a directed graph '$G = (V, E)$'. Here 'V' represents the set of radio nodes (i.e. nodes) and 'E' represents the pair of links '(i, j)' between two radio nodes 'i' and 'j'. For a comprehensive 'T', link quality metrics where 'T' represents the frequency of constraints assumed in our work, there refers to a link quality of the '$T\,th$' metric, with '$w_T(i, j)$, $t = 1, 2, \ldots, T$' between two nodes 'i' and 'j'.

A path, '$P = \{(S, i), (i, j), \ldots., (m, D)\}$', represents a succession of links associating between a source 'S', and destination 'D', respectively. Furthermore, a transmission quality '$w(P)$' for '$T\,th$' metrics '$w_T(P)$' represents the sum of all correlated link quality values for '$T\,th$' metric. Then, a concave aggregation function with a concave metric such as link stability 'LS', a bottleneck link between nodes 'i' and 'j' is directly concerned to it end-to-end delay '$E2ED$'. Hence, the

Fig. 1 Structure of Mesh Routing Table

Origin Qualifier (OQ)	Target Qualifier (TQ)	Origin Series Number (OSN)	Target Series Number (TSN)	Telecast Qualifier (TQ)	Time to Hangout (TTH)

aggregated function for a concave metric with a minimum of all individual link quality is expressed as given below.

$$w_T(P) = MIN\ \{(i,j) \in P \mid w_T(i,j)\} \tag{1}$$

Then, from the above Eq. (1), a constraint for the 'T th' metric '$C_T \geq 0$' is represented as given below.

$$C = \{c_T | 1 \leq t \leq T\ \} \tag{2}$$

Then, the objective of diversified-control routing model is to identify optimal route that make certain all '$c_T \in C$'. Then evaluation function is represented as given below.

$$EF_t(P, c_T) = \{[w_T(P) \geq c_T]\} \tag{3}$$

The above evaluation function, 'EF' results, in either '0', or '1'. If the result of the evaluation function is '1', then the pathway 'P' is said to be achievable for the control 'c_T'. On contrary, if the result of the evaluation function is '1' for all 't', then, the pathway is feasible for all the assumed controls. With the above network and system model, the elaborate description of the MST-OLSC method is given below.

For flexible sensor scheduling, Markov State Transition model is adopted to classify the IoT-enabled WMN into three different statuses such as, active, inactive and broadcast. In the proposed model, each sensor operating model operates under different bandwidth (i.e. data transfer rate across a given path). In other words, entire range of bandwidth is said to be consumed only by the active mode, the sensor in the inactive mode preserves the bandwidth for future purpose and specifically, the sensor ready for broadcasting consumes higher bandwidth. To follow a different pattern of optimal bandwidth allocation, the network obtains the required bandwidth and potentiality of each sensor. Then, operating design is applied to bandwidth which is paired for reducing the bandwidth consumption.

3.4 Markov state transition scheduling model

In this proposal, Markov State Transition Scheduling (MSTS) model is utilized for IoT-enabled WMN. The sensor node status are defined from three operating modes, as active, inactive and broadcast respectively. To make things simple, one mode is finite for each sensor at a given time period and based on the state, the sensor is said to be scheduled accordingly in the next time interval. The node is structuring said that designed in a clockwise patter, that a sensor with higher bandwidth only has the potentiality to transfer to a lower bandwidth stage and not in the anticlockwise pattern. Specifically, the Markov state transition scheduling model is utilized to obtain the probabilistic node state transitions according to the constraints of the available tokens in the buffer.

Also to reduce the congestion occurring due to many sensors in WMN, additional optimization of overall network bandwidth is performed by stopping the propagation of packets upon the detection of inactive mode. In this manner, network effectiveness is attained in terms of both bandwidth and network traffic saving in IoT. Fig. 2 given below shows the flow diagram of Markov State Transition Scheduling Model.

As illustrated above, to reduce an occurrence of congestion in IoT-enabled WMN, a Markov State Transition Scheduling model is designed for reducing the number of transmitted

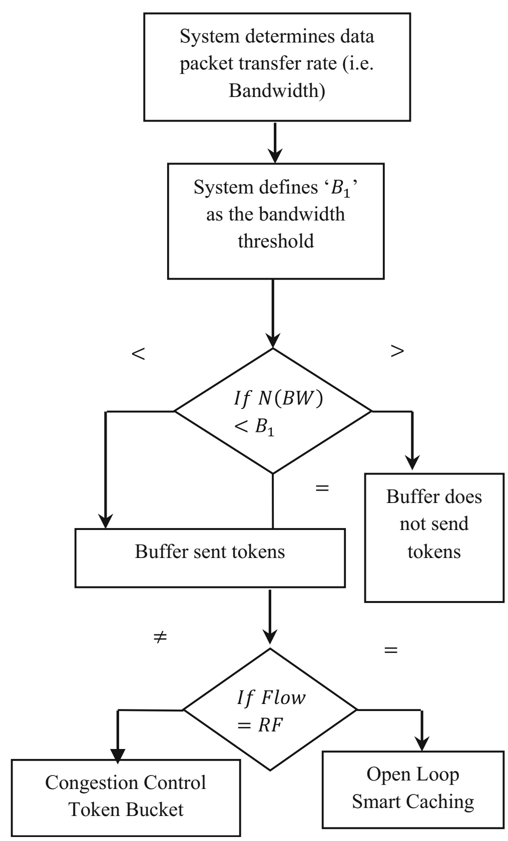

Fig. 2 Flow diagram of Flexi Variable Markov Model

interest packets. With this objective, 'B_1' is initialized as the threshold value of bandwidth or the data transfer rate by the system. If the system observes that the bandwidth level of sensor is greater than 'B_1', the buffer does not send the tokens to the sensor due to larger number of packets arrival in IoT-enabled WMN. When the bandwidth level of sensor is found to be smaller or equal to 'B_1', the buffer send the tokens to the sensor for congestion control. The pseudo code representation of Markov network based Estimation of State Transition is given below.

Algorithm 1 Markov network based Estimation of State Transition

Input: Origin Qualifier, 'OQ', Target Qualifier 'TQ',
Origin Series Number 'OSN', Target Series N 'TSN', Telecast Qualifier 'TQ', Buffer '$Buff$', Tokens 'Tns', Bandwidth Threshold 'B_1'
Output: Bandwidth-oriented Congestion Avoidance
Step 1: **Begin** Step 2: F**or** each Origin Qualifier, 'OQ', sequence number 'SSN', with specified destination identifier 'DID', sequence number 'DSN' Step 3: Obtain diversified-control routing using (2) and (3) Step 4: Measure available and probable states using (4) and (5) Step 5:**If** node bandwidth > bandwidth threshold Step 6: Buffer does not sends tokens to host computer Step 7: **End if** Step 8: **If** node bandwidth < B_1 Step 9: Buffer sends tokens to host computer Step 10: Call Open Loop Smart Caching function Step 11: **End if** Step 12: **If** node bandwidth >= B_1 Step 13: Go to step 3 Step 14: If node bandwidth <≠ B_1 Step 15: Call Congestion Control Token Bucket Caching Step 16: **End if** Step 17: E**nd for** Step 18: **End**

As given in the above Markov network based on Estimation of State Transition algorithm, the idea of different states concept is adopted for designing transition pattern of radio nodes as IoT devices pertaining to different corresponding bandwidth states. In this work to minimize the energy consumption, a State Betweenness Centrality theorem is used to measure the centrality based on conditional distributions. Let '$P = \{P_1, P_2, \ldots, P_n\}$' represent a group of random variables 'P_i' in which '$p_i D$' denotes a discrete set or a random set. Let '$Prob(P_i = p_i)$' be the probability that '$P_i = p_i$'. Then, the probability '$Prob(P_i = p_i)$' and the joint probability '$Prob(P_i = p_i, \ldots ., P_n = p_n)$' is said to be concatenated as'$Prob(P)$'. The condition that 'P' is the MSTS of a node 'N' in the set of a sensor's all available states '$States$' then defined as given below.

$$Prob\ (p_i|States), States \in Tns \tag{4}$$

$$Prob(p_i|States) = Prob(p_i|\ P_{Ni}) \tag{5}$$

From the above Eq. (4), 'Tns' represent the availability of tokens in the buffer. MSTS model attempt to measure the conditional distributions of the form '$Prob(p_i|\ States)$' where the context length '$|States| \leq Tns$' differs according to the available tokens 'Tns' in the buffer '$Buff$'. In Hierarchical RNN [1], balanced routing was not performed due to the lack of determining the context length. This in turn increased the energy consumption during routing and therefore causing end-to-end delay. This drawback is overcome by the proposed method with the utilization of State Betweenness Centrality theorem.

3.5 Open loop smart caching model

In this section, an Open Loop Smart Caching model for congestion control and Open Loop Smart Caching model is designed with the objective of increasing the packet delivery ratio in IoT-enabled WMN. This is called as Open Loop Smart Caching as it tries to prevent congestion with the objective of making sure that the network can carry the offered data packet. It is achieved by following two policies. First, according to the availability of tokens in the buffer, data packet is received from the host computer. In other words, the host computer throws the data packets when the bucket is full. On the other hand, with the constant data transfer rate, end-to-end delay between the nodes is reduced. Here, the data packet was discarded when the bucket is full.

Figure 3 shows the flow diagram of Open Loop Smart Caching model. As shown in the figure, the left side represents the congestion control mechanism by means of the traffic policing and the right side represents the flow control mechanism by means of traffic shaping.

The pseudo code representation of the Integrated Congestion and Flow Controller is given below.

Algorithm 2 Integrated Congestion and Flow Controller

Input: Origin Qualifier 'OQ', Target Qualifier 'TQ', Buffer '$Buff$', Tokens 'Tns', Data Packets '$DP = \{DP_1, DP_2, \ldots, DP_n\}$', data transfer rate '$\Delta R$',
Output: Congestion Control

Step 1: **Initialize** Buffer '$Buff$', Available Tokens 'Tns', Data Transfer Rate 'DTR'
Step 2: **Begin**
Step 3: **For** each Origin Qualifier 'OQ' with Target Qualifier 'TQ' and data transfer rate 'ΔR'
//addresses overflow
Step 4: **If** '$n(DP) < Avl\ (Tns)$' **then**
Step 5: Sent data packets between host node and destination node
Step 6: **End if**
Step 7: **If** '$n(DP) > Avl\ (Tns)$' **then**
Step 8: Don't sent data packets between host node and destination node
Step 9: **End if**
//addresses data constant rate
Step 10: **If** '$(DTR) < \Delta R$ '
Step 11: Constant data flow
Step 12: **End if**
Step 13: **If** '$(DTR) > \Delta R$ '
Step 14: Stored data packets in buffer
Step 15: Sent data packet at constant interval
Step 16: Constant data flow
Step 17: **End if**
Step 18: **End for**
Step 19:**End**

The Congestion Control Traffic Policing does allow saving up to maximum size of the bucket 'n'. Next, as shown in the figure, there is a presence of bucket or buffer '*Buff*'. Multiple tokens '*Tns*' are present in the bucket or buffer. The data packets '$DP = \{DP_1, DP_2, \ldots, DP_n\}$' get in to the bucket or buffer from the host computer '*HC*'. One token is added to the

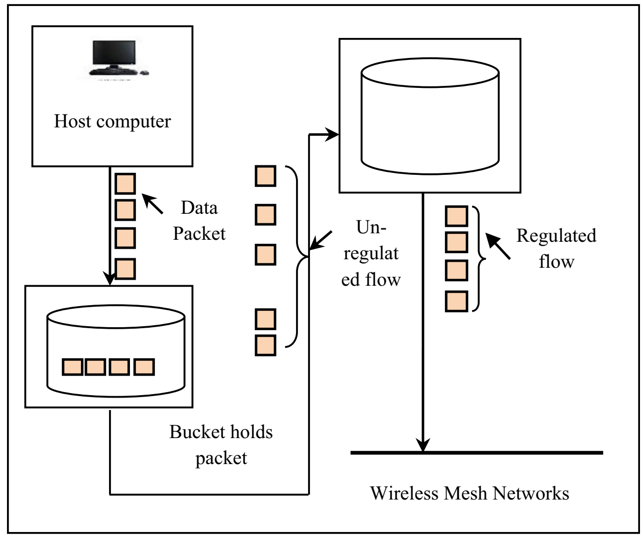

Fig. 3 Open Loop Smart Caching model

bucket at the rate of 'ΔR' (i.e. data transfer rate or data packet transfer rate) and the bucket or buffer holds the token. Then, depending on the number of available tokens in the token bucket or token buffer, that number of frames or data packets comes inside. Therefore, no scope of overflow is said to occur.

So the token bucket prohibits the situation from overflow. Hence, depending on the free tokens available in the bucket or buffer, only that number of packets will comes inside the buffer or bucket. However, the token bucket does not ensure that the inflow data packet or data rate are said to be obtained in the outflow. So, in this work, a novel Open Loop Smart Caching mechanism is presented. To ensure constant data rates or packets which are getting injected into the network, a Traffic Shaping Flow Control mechanism is designed. Figure 4 given below shows the inflow and outflow of a smart caching bucket.

Constant data rates are said to be ensured in our work by means of smart caching. Therefore, the design of our method is the presence of a host computer, followed by which a traffic policing algorithm is designed that prohibits situation from overflow, a traffic shaping algorithm that ensures constant data packets being injected into networks and finally, the networks as shown in the figure.

As shown in Fig. 4, x axis represents the time taken to transfer the data or the data transfer rate and the y axis represents the data packets to be transmitted in the network. From Fig. 4(a), input to a smart caching bucket is being represented as the '25 $MB/\sec$ *for* 40 ms' with the application of smart caching bucket to ensure constant data rate. The data packets are stored in buffer and then send at regular intervals with the mean rate of '10 $MB/\sec$ *for* 100 ms' as shown in Fig. 4(b). Hence, with the application of smart caching bucket, the x axis represents the time taken to transfer the data or the data transfer rate represented in terms of milliseconds (ms). The y axis represents the actual data packets to be transmitted. In this

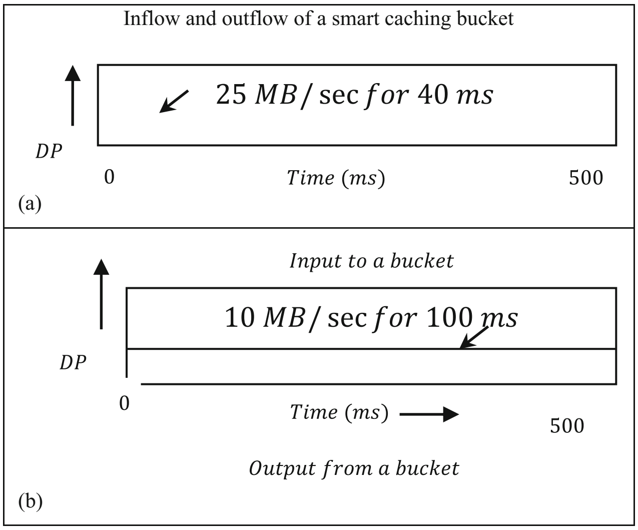

Fig. 4 Data flow of sample input and output smart caching bucket

way, the input flow is said to be equated to the output flow. In this manner, end-to-end delay said to be reduced at constant data rate.

4 Experimental result

The effectiveness of the proposed congestion control method is discussed.

4.1 Experimental environment

The proposed method is compared with the existing Hierarchical RNN [1] and Cognitive Heterogeneous Routing (CHR) using three constraints: energy consumption, end-to-end delay and packet delivery rate. The proposed method and the existing work were evaluated using NS-3 Network Simulator [20]. Based on the usage of WMNs, mesh routers are connected with each other with a bandwidth rate of 11 Mbps and delay including 2 ms and the wireless channels transmission range is assigned to 140 m. The configuration for simulations is provided in Table 1.

In the two-dimensional area of 1000 m * 1000 m, 40, mesh routers are placed in a uniform distribution pattern. Three different network topologies were generated to obtain an average result with the same data point at 10 different simulation runs. Based on the transmission range, each mesh router is connected to its neighbor mesh routers with corresponding values stored in the routing table. For mesh clients, node mobility utilizing the random waypoint model is applied with a speed between zero to ten milliseconds and a 15 s pause time. Using a random destination sets, multiple application flows are utilized, in such a manner that the destination nodes were different from their source nodes via source sequence and destination sequence number acquired from the routing table.

For experiments, IoT-enabled sensors are utilized that possess three QoS constraints: link stability, bandwidth and data flow rate. Data packet size was assumed to be 1024 bytes. The simulation time set to 100 s. 10 repetitions (i.e. 10 different simulation runs) are performed for each individual setting of simulation and average their results.

4.2 Experimental analysis and results

The efficiency and effectiveness of the MST-OLSC method was compared with two different existing methods, Hierarchical RNN [1] and CHR [2] in terms of energy consumption, packet delivery rate and end-to-end delay.

4.2.1 Evaluation of energy consumption

The consumption of energy denotes the energy required or is spent by the nodes to perform a stipulated task. In this work, the stipulated task control the congestion for IoT-enabled WMN within the simulation time. This energy consumption is arrived at by measuring each IoT-enabled node's energy at the stopping point of the simulation, factoring in the initial energy of each node. In other words, the energy consumption refers to the energy consumed in attaining congestion control mechanism. The following formula will produce the value for energy consumption:

$$EC = \sum_{i=1}^{n} N_i * EC\,[CC] \tag{6}$$

From the above Eq. (6), the energy consumption 'EC', is measured by means of the number of IoT-enabled nodes considered for experimentation 'N_i' and the energy consumed in controlling the congestion '$EC\,[CC]$'. It is measured in terms of joules (J). Table 2 given shows the energy consumption values for three different methods.

Figure 5 given below shows the graphical representation of energy consumption with respect to 500 different nodes considered in the x axis, measured in terms of joules. Number of nodes is directly proportional to the energy consumption. In other words, by increasing nodes, data packets ready for transmission also increases and consequently, energy consumption is also found to be increasing.

In this evaluation, energy consumption means the energy being consumed while controlling the traffic between nodes. The results show that our MST-OLSC methods show lesser energy being consumed than other congestion control methods, Hierarchical RNN [1] and CHR [2] respectively. Especially, the proposed MST-OLSC method reduces the energy consumption when the network scale gets smaller. From the simulation conducted with 50 nodes, the energy consumed for congestion control using MST-OLSC was found to be

Table 1 Sample configuration for simulations

Parameters	Values
Network simulator	NS 3.26
Time for simulation	100 s
Area concerned for simulation	1000 m * 1000 m
Frequency of router mesh	25
Frequency of client mesh	20
Distribution of node in network	Uniform random
Transmission range	140 m
Mobility model	Random waypoint
Link propagation delay	2 ms
Packet size	1024bytes
Number of nodes	50, 100, 150, 200, 250, 300, 350, 400, 450, 500

'1.75 J', '2.1 J' using [1] and '2.4 J' using [2]. With this, the comparative analysis shows betterment achieved with MST-OLSC because of the incorporation of Estimation of State Transition that effectively classifies the radio node status accordingly to three different operating modes. In addition with the State Betweenness Centrality based on the measure of centrality, conditional distributions are made and flexible sensor scheduling is ensured. With this flexible sensor scheduling, the energy being consumed in controlling the traffic has to be reduced by 23% when compared to [1] and 34% when compared to [2] respectively. On the other hand, in [1, 2] the aspects involved in energy consumption was not analyzed. With the above results it is inferred that the MST-OLSC method results in low packet drop rate, therefore reducing the congestion.

4.2.2 Evaluation of packet delivery rate

The second factor to be analyzed is the packet delivery rate. It denotes the percentage ratio of data packets that were received at the target place to the data packets that are initiated by the origin qualifier. This parameter ensures the discharge of data packets from origin qualifier to target node. Higher ratio ensures superior achievement of routing efficiency.

$$PDR = \frac{DP_R}{DP_S} * 100 \tag{7}$$

From the above Eq. (7), the packet delivery rate '*PDR*', is obtained based on the number of packets received 'DP_R' and the number of packets sent 'DP_S'. It is measured in terms of percentage (%).Table 3 given shows the packet delivery rate values for three different methods.

Fig. 6 above depicts the packet delivery rate of our proposed MST-OLSC method, existing Hierarchical RNN [1] and CHR [2] in accordance with the various number of data packets. As depicted in the figure, increasing the number of data packets, higher packets gets

Table 2 Energy consumption using MST-OLSC, Hierarchical RNN [1] and CHR [2]

Number of nodes	Energy consumption (J)		
	MST-OLSC	Hierarchical RNN	CHR
50	1.75	2.1	2.4
100	1.95	2.55	2.75
150	2.05	2.85	3.35
200	2.19	3.15	3.95
250	2.25	3.55	4.05
300	2.48	3.85	4.35
350	3.55	4.15	4.55
400	4.85	5.35	5.85
450	5.35	5.95	7.35
500	7.25	9.25	12.55

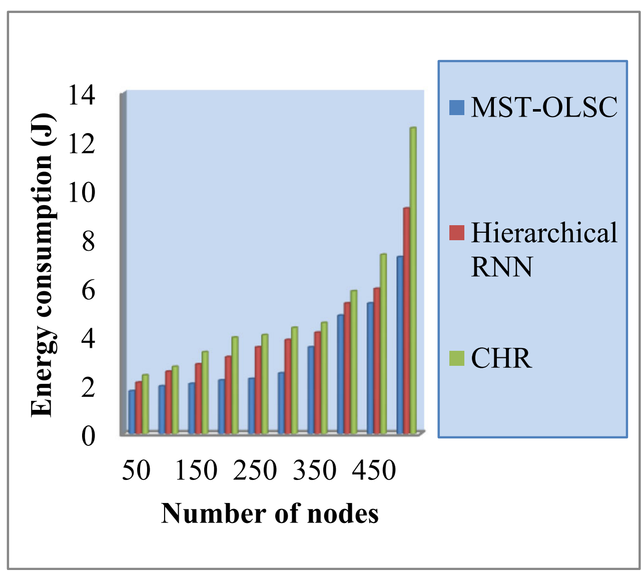

Fig. 5 Performance graph of energy consumption

accumulated in the buffer and accordingly congestion has to be handled. Therefore, the data packets are found to be inversely proportional to the packet delivery rate. In other words, higher the data packets, considerably lesser the packet delivery rate is. From the simulation conducted with 15 data packets, it was found that '14' data packets were received at the destination using MST-OLSC, '12' data packets were received at the destination using [1] and '11' data packets received at the destination using [2]. With this, the packet delivery rate were observed to be '93.33%', '80%' and '73.33%' using MST-OLSC, [1, 2] respectively. With this, the evaluation results show that the MST-OLSC method can gain higher congestion control factor than [1, 2]. This is because the MST-OLSC method integrates both the measures to be addressed in case of overflow and data constant rate by means of Open Loop Smart Caching mechanism. By applying this integrated algorithm, both the measures are said to be handled and therefore safeguarding higher packet delivery rate. On the other hand, in [1] a probability of exploratory packet being identified with a successful path may result in imbalance between the exploitation and exploration problem, causing minimum packet delivery rate. In this case MST-OLSC exploitation and exploration problem is addressed by mans of Open Loop Smart Caching whereas well balanced isolation between the packets in bucket is ensured, therefore contributing to packet delivery rate. In a similar manner in [2] exploration was ensured where the parameters were initialized in this stage and only calculates the probability of data transmission using Q Learning. With this, the packet delivery rate using MST-OLSC method have found to be better by14% when compared to [1] and 27% when compared to [2] respectively.

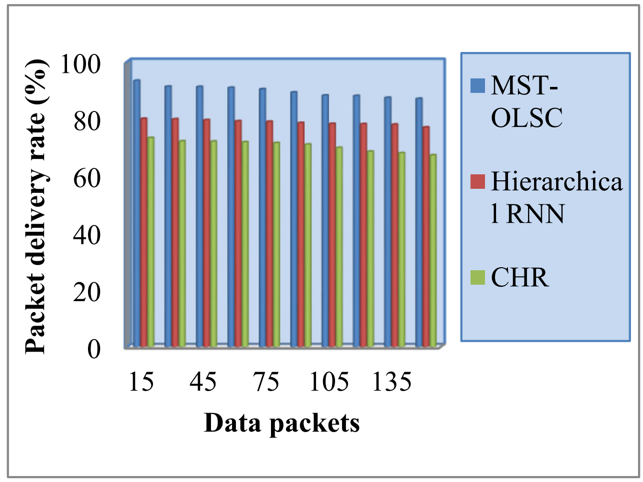

Fig. 6 Performance graph of packet delivery rate

4.2.3 Evaluation of end-to-end delay

Finally, the third factor to be analyzed is the end-to-end delay that refers to the time consumed for data packets sent by the sensor be a propagated along the network (i.e. IoT-enabled WMN) from a source node to destination node. It is measured as given below.

$$E2ED = \sum_{i=1}^{n} DP_i * Time\ [S \rightarrow D] \tag{8}$$

From the above Eq. (8), the end-to-end delay '$E2ED$' is measured based on the frequency of packets considered for experimentation 'DP_i' and the time consumed in propagating packets between the source and destination '$Time\ [S \rightarrow D]$' respectively. It was evaluated with reference to milliseconds (ms). Table 4 given shows the end-to-end delay values for three different methods.

Table 3 Packet delivery rate using MST-OLSC, Hierarchical RNN [1] and CHR [2]

Data packets	Packet delivery rate (%)		
	MST-OLSC	Hierarchical RNN	CHR
15	93.33	80	73.33
30	91.25	79.85	72.15
45	91.17	79.55	72.05
60	90.85	79.15	71.85
75	90.35	79	71.55
90	89.25	78.55	71
105	88.15	78.25	69.85
120	88	78.15	68.55
135	87.35	78	68
150	87	77	67.25

Table 4 End-to-end delay using MST-OLSC, Hierarchical RNN [1] and CHR [2]

Data packets	End-to-end delay (ms)		
	MST-OLSC	Hierarchical RNN	CHR
15	0.525	0.63	0.825
30	0.585	0.685	0.855
45	0.615	0.725	0.885
60	0.635	0.745	0.935
75	0.655	0.785	0.955
90	0.715	0.815	0.975
105	0.735	0.855	0.995
120	0.785	0.875	1.025
135	0.815	0.925	1.055
150	0.845	0.945	1.085

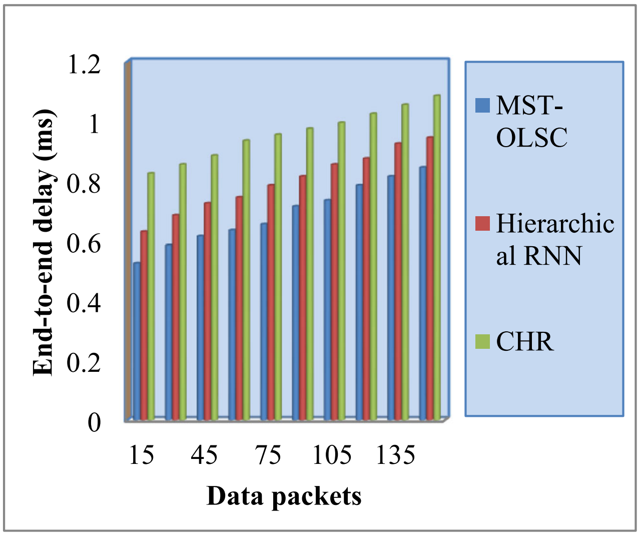

Fig. 7 Performance graph of end-to-end delay

Fig. 7 given above illustrates the end-to-end delay provided in controlling the congestion using the proposed MST-OLSC and the existing Hierarchical RNN [1] and CHR [2] respectively. From the figure it is inferred that increasing the data packets causes consideration of higher IoT-enabled nodes in the network. With this there arises a significant increase in the end-to-end delay also. This is evident from the sample simulation conducted with 15 data packets where the end-to-end delay was observed to be '0.525 ms' using MST-OLSC, '0.63 ms' using [1] and '0.825 ms' using [2] respectively. From the simulations it is inferred that the MST-OLSC method performs comparatively better than the other two methods in terms of the average end-to-end delay. The performance improvement gets more apparent as the data packets escalates, where the time consumed for packets in traversing between origin and target node escalates. As mentioned earlier, MST-OLSC method can consider the end-to-end path situation for every data packet. However, Hierarchical RNN [1] schedules data packets based on epoch training, while theCHR [2] schedules data packets based on frequency bands without considering the end-to-endpath situation. As a result, both [1, 2] are more sensitive than MST-OLSC method to the number of hops along the path. This in turn to degrade its delay performance by 14% compared to [1] and 28% compared to [2] respectively.

5 Conclusions

A highly efficient congestion control mechanism for IoT-enabled WMN is designed. To start with, a new Markov State Transition Scheduling model is designed to differentiate between three different statuses. Next, a novel Markov network based Estimation of State Transition along with State Betweenness Centrality with conditional distributions is applied to minimize the energy consumption. Open Loop Smart Caching algorithm is designed, to solve the congestion as addressing overflow and provide a constant data rate under three different constraints, link stability, bandwidth and data flow rate for IoT-enabled WMN. The overflow issues are addressed, isolation and communication between nodes in network is ensured, thereby improving the packet delivery rate and consequently reducing the end-to-end delay. With this, a constant data rates flowing in the input and output is said to be ensured, this results in the overall network minimizes traffic load, and in turn energy consumption. The simulation results showed that the integrated method performs better with improved packet delivery rate and energy consumption, as well as reduced end-to-end delay for IoT-enabled WMNs.

References

1. ArturRataj, "Random Neural Networks with Hierarchical Committees for Improved Routing in Wireless Mesh Networks with Interference", Springer Nature Computer Science, Oct 2019 – Hierarchical RNN
2. Ahmed Al-Saadi, Rossitza Setchi, Yulia Hicks, Stuart M. Allen," Routing Protocol for Heterogeneous Wireless Mesh Networks", IEEE Transactions on Vehicular Technology, Vol. 65, No. 12, Dec 2016 – Cognitive Heterogeneous Routing (CHR)
3. Armir Bujari, Andrea Marin, Claudio E. Palazzi,Sabina Rossi, "Smart-RED: a novel congestion control mechanism forHigh throughput and low queuing delay", Wireless Communications and Mobile Computing, Wiley, 2019
4. Yuvraj Sahni, Jiannong Cao, Shigeng Zhang, Lei Yang, "edge mesh: a new paradigm to EnableDistributed intelligence in internet of things", IEEE Access, Sep 2017
5. Junseok Kim, Seongwon Kim, Tarik Taleb, Sunghyun Choi, "RAPID:contention resolution-based random Accessusing context ID for IoT", IEEE Transactions on Vehicular Technology, Jul 2015
6. JongGwan An, Wenbin Li, Franck Le Gall, Ernoe Kovac, Jaeho Kim, Tarik Taleb, Jae Seung Song, "EiF: toward elastic IoT fog framework for AIServices", IEEE Communications Magazine, 2019
7. Hichem Sedjelmaci, Sidi Mohamed Senouci, Tarik Taleb, "An Accurate Security Game for Low-Resource IoT Devices", IEEE Transactions on Vehicular Tecnology, Vol. 6, No. 10, 2017
8. Wei Li, Fan Zhou, Kaushik Chowdhury, Waleed Meleis, "QTCP: Adaptive Congestion Control with Reinforcement Learning", IEEE Transactions on Network Science and Engineering (Volume: 6, Issue: 3, 2019)
9. N. Akkari, P. Wang, J. M. Jornet, E. Fadel, L. Elrefaei, M. G. A. Malik, S. Almasri, I. F. Akyildiz, "Distributed timely-throughput optimal scheduling for the internet of Nano-things", IEEE Internet of Things Journal (volume: 3 , Issue: 6 , 2016)
10. Juan Pablo Astudillo León, Luis J. de la Cruz Llopis, "Emergency aware congestion control for smart grid neighborhood area networks", Ad Hoc Networks, Elsevier, 2019
11. Wonyong Yoon, Dongman Lee, Byoungheon Shin, SeonYeong Han, "Price-based congestion control and local channel-link assignment for multi-radio wireless mesh networks", Computers and Electrical Engineering, Elsevier, 2013

12. Simone Bolettieri, GiacomoTanganelli, Carlo Vallati, EnzoMingozzi, "pCoCoA: a precise congestion control algorithm for CoAP", Ad Hoc Networks, Elsevier, 2018
13. Md. EmdadulHaque, Faisal Tariq, Laurence Dooley, Ben Allen, Yan Sun, "Efficient congestion minimisation by successive load shifting in multilayer wireless networks", Computers and Electrical Engineering, Elsevier, 2018
14. Maheen Islam, Md. AbdurRazzaque, Md. Mamun-Or-Rashid, Mohammad Mehedi Hassan, Ahmad Almogren, Abdulhameed Alelaiwi, "Dynamic traffic engineering for high-throughput data forwarding in wireless mesh networks", Computers and Electrical Engineering, Elsevier, 2016
15. JieJia, Qiusi Lin, Jian Chen, Chunyu Li, Xingwei Wang, "Congestion aware channel allocation with route scheduling in wireless cognitive radio mesh network", Computers and Electrical Engineering, Elsevier, 2013
16. Ihsan Ayyub Qazi, Taieb Znati, "On the design of load factor based congestion control protocols for next-generation networks", Computer Networks, Elsevier, Jun 2011
17. Weiqi Chen Quansheng Guan, Shengming Jiang, Quanxue Guan, Tiancheng Huang, "Joint QoS provisioning and congestion control for multi-hop wireless networks", EURASIP journal on wireless communication and networking, Springer, 2016
18. Luís Barreto, "XCP-Winf and RCP-Winf: Improving Explicit Wireless Congestion Control", Journal of Computer Networks and Communications, Hindawi, 2014
19. Adel A. Ahmed Waleed Ali, "A lightweight reliability mechanism proposed for datagram congestion control protocol over wireless multimedia sensor networks", Wiley, 2018
20. "NS-3 simulator, version 3.26," 2018, https://www.nsnam.org/releases/ns-allinone-3.26.tar.bz2

N. Yuvaraj received his Bachelor degree in B.Tech. Information Technology and Master degree in M.E. Computer Science and Engineering from SSM College of Engineering, Komarapalayam. He has completed his doctorate from Anna University, Chennai. He has more than 7 years of teaching experience from reputed Engineering colleges and currently working as an Assistant Professor in Department of Information Technology at J.K.K. Nattraja College of Engineering and Technology, Komarapalayam. He is a member of MISTE, and IFERP. His current research interests are primarily in Networking, Data Mining, Cloud Computing.

G. Saravanan received his M.E. degree in Computer Science and Engineering from Government College of Technology, Coimbatore, Anna University, Chennai. He obtained his doctorate from Anna University, Chennai. He has 12 years of teaching experience. He is a member of MISTE, IAENG and IFERP. At present, G.Saravanan is working as an Assistant Professor in the Department of Computer Science and Engineering at Erode Sengunthar Engineering College, Perundurai, Tamilnadu, India.