Received: March 25, 2024. Revised: May 10, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Efficient Congestion Control Mechanism and Handover Management Scheme for Performance Enhancement of TCP over IoT Network

Sultana Parween[1]*

[1]*Department of Computer Science & Engineering, Jamia Hamdard, New Delhi, India*
* Corresponding author's Email: sultanaparween@jamiahamdard.ac.in

Abstract: Transmission Control Protocol (TCP) used multipath for achievement of data instantaneously to improve its performance. Anyhow, the preceding TCP protocol in the Internet of Things (IoT) network experienced a struggle to transmit a superior number of subflows. The Bottleneck Bandwidth and Round-trip propagation time (BBR) generate a network path model by assessing the available bottleneck bandwidth and the minimal Round-Trip Time (RTT) to achieve the desired delivery rate, consequently reducing the latency. However, some research studies have indicated the presence of significant fairness issues related to RTT in the BBR algorithm. To overcome this issue, we proposed a New Enhanced Congestion Control (CC) mechanism referred to as Bottleneck Bandwidth and Round-trip propagation time (BBR-NEnh) which is cross-layer-based efficient congestion control in a 5G IoT heterogeneous network. The proposed work encloses three sequential processes including grid-based network construction, improved two-factor clustering, and Hybrid TCP congestion control. Initially, we perform network construction by grid-based to improve the information transfer rate and increase connectivity among devices. After that, IoT devices are clustered based on two factors congestion and buffer space using the Improved K-Medoids algorithm (IK-Med) by the edge server. Further, to enhance the efficiency of TCP, the proposed research adopts a BBR-NEnh by optimizing the congestion control parameters using the Adaptive Tunicate Swarm Optimization (ATSO) algorithm. Finally, the dynamic handover and adjustment of windows are performed based on the Fast Adaptation Technique (FAdT) which can send the data quickly without congestion. The implementation of the proposed research is carried out using NS-3.26 and the performance of the proposed BBR-NEnh model is evaluated using a variety of performance metrics, including goodput, delay, packet loss, queue length, transmission rate, and throughput. The performance of the proposed work is compared with IMPRTT (Improved RTT) and CW-IoT (Congestion Window Algorithm for the Internet of Things). The average throughput, goodput, and packet loss of the proposed work for 1500 packets size are 4, 0.9, and 10.3, and the average delay, transmission rate, and queue length over 150 seconds are 0.004, 2500, and 36.1, demonstrating that the proposed work outperforms earlier work like IMPRTT and CW-IoT.

Keywords: Internet of things (IoT), 5G, Congestion control (CC), Transmission control protocol (TCP).

1. Introduction

The number of user equipment is massively increasing day by day, so small cells are densely deployed in a 5G heterogeneous network to give the best coverage and to improve network performance. Sensor nodes are utilized to collect data and transmit it to networked IoT devices. These devices are then, employed for a variety of applications, including but not limited to healthcare and home automation. Numerous amounts of applications are in the 5G network and it has high complexity. For complexity reduction in the 5G network, proper network construction and management is necessary. For data transmission between heterogeneous IoT devices in a 5G network, many IoT protocols such as MQTT (Message Queuing Telemetry Transport), AMQP (Advanced Message Queuing Protocol), and CoAP (Constrained Application Protocol) are used for large-scale communication. However, there are several challenges associated with IoT data transmission between sensor nodes. One of these challenges is the high transmission rate, which often results in significant network congestion. In order to overcome these challenges, the TCP has been introduced to facilitate efficient data transmission. It is specifically designed to handle the huge amount of data traffic that is transmitted globally. It is also capable of managing network congestion in the IoT network, thereby improving the overall QoS. It offers efficient data transmission with superior performance with respect to load, connectivity, and speed compared to UDP (user datagram protocol). The TCP protocol faces various challenges that impact data transmission, such as fixed RTT and Retransmission Time Out (RTO), high buffer rate, transmission delay, and bandwidth consumption. These challenges contribute to increased data congestion. Various methods have been introduced to manage congestion in TCP. The above existing issues are resolved by our proposed work using improved network construction, advanced algorithms, and techniques.

1.1 Objectives & motivation

This research aims to overcome the challenges associated with excessive bandwidth consumption, prolonged transmission latency, substantial packet loss, and insufficient throughput with the intention of enhancing the efficiency of TCP.

•**Improper network construction and management:** In prior studies, IoT device deployments were improperly managed, resulting in high packet loss ratios, improper CC, and computational complexities.

•**Unnecessary traffic:** In some existing works, missing packets, duplicate packets, and negative acknowledged packets remain in the network, which leads to high complexity, unnecessary traffic, and degrades the performance of TCP.

•**Inefficient CC:** Previous research on CC has primarily focused on utilizing RTT and RTO information. However, these approaches fail to accurately estimate congestion levels based on the current state of the network, resulting in significant packet loss and transmission delays.

This research work aims to achieve the following objectives:

- To further, reduce the complexity of TCP through enhanced network construction that mitigates the substantial packet loss ratio and processing complexities.
- In order to intellectually manage the entities of the network through the implementation of clustering, which enhances energy efficiency, mitigates imbalanced load, and reduces computational complexity.
- To enhance the efficiency of fast retransmission by implementing handover, which resolves the issues of packet reordering and spurious retransmission timeout.
- To improve the TCP performance by implementing a cross-layer-based CC mechanism in which congestion is predicted and notified. Further, predicted congestion is reduced which increases the performance of TCP.

1.2 Research contribution

The following are the primary contributions of the research:

- To enhance network system, grid-based network construction is suggested, which decreases the complexity in TCP performance.
- Using IK-Med, edge servers execute better two-factor clustering to reduce high complexity. This lengthens the lifespan of the network device thereby decreasing processing time and computational complexity.
- The BBR-NEnh method is proposed with adaptive tunicate swarm optimization (ATSO) by taking into account many efficient factors in order to improve congestion control and avoidance. Additionally, BBR-NEnh anticipates and warns of congestion, which enhances TCP performance by avoiding congestion.
- Packet handover is based on the time interval between two ACK (Acknowledgement) messages sent while conducting fast retransmission, which eliminates the issue of packet reordering and erroneous retransmission timeout.

1.3 Organization of the paper

The following outline represents the structure of this research paper:

- Related works in congestion control in TCP and research gaps (Section 2).
- Problem statement (Section 3).
- Proposed research work (Section 4)
- Simulation setup, Performance analysis, and Comparative analysis (Section 5).

Table 1. Notation

Notation	**Description**
ϖ	Grid size
L_i and L_j	Clustered nodes for transmission
$\vec{D}_i$ and $\overrightarrow{D_j}$	Medoids of the ith and jth cluster.
∂_{ecn}	ECN mark rate
ξ	Bottleneck link
$\vec{Y}_o^{min}$	Location of the node
b_1	Arrangement of the node ranges
ACK_0	Duplicate acknowledgement

- Conclusion (Section 6).

Table 1 represents the notation form of the proposed work.

2. Related works

This section deals with the survey of literature on efficient handover and CC for TCP in 5G IoT networks. The authors proposed an SRQ-PI congestion tracking scheme for efficient control of congestion [1]. The estimation of the incoming traffic rate was conducted using a self-tuning rate and queue-based proportional and integral controller (SRQ-PI) that relied on the measurement of queue length. Simulation results demonstrate that SRQ-PI was stable, had faster transmission, lower jitter, and was robust against dynamic network parameters. This pays less attention to SRQ-PI tunes itself and then adjusts the control gain parameters based on feedback. However, it cannot efficiently predict and control the congestion, which leads to improper CC and reduces the performance of TCP. The authors introduced a modified BBR algorithm for CC in [2]. The BBR-S algorithm incorporates the Adaptive Tobit Kalman Filter (ATKF) as a replacement for the max filter, mitigating the problem of overestimation. This work pays less attention to the fairness gain of the protocol was improved by using BBR. The BBRV2 CC algorithm was proposed for data transmission [3]. Here, BBRV2 consists of four processes such as startup, drain, ProbeBW, and ProbeRTT. In the start-up phase, the pacing rate and CWND size should be increased. After the start-up phase, in the drain phase continuous bandwidth detector reaches a stable value and marks their respective threshold rate of CWND. After that, Probe cycles increase the inter-protocol fairness of RTT and transfer the data by reducing the congestion in CWND. Finally, BBRV2 is a promising choice for high-speed networking with a short queue. However not taking RTO into account leads to a decrease in throughput. TCP was suggested as a way to recover data in multi-radio dual connectivity for handover [4]. The authors in [5] presented an enhanced TCP CC algorithm designed for the next generation. In this context, several variants are suggested to enhance the efficiency of TCP in diverse scenarios, particularly in networks characterized by loss and high bandwidth-delay product. Finally, evaluation results demonstrate that improved TCP achieves high goodput and fairness and reduces CC. However, this approach lacks the inclusion of crucial criteria such as RTO, RTT, and queue status. In [6], the authors proposed an algorithm that is used to estimate RTO for reducing delay and improving the convergence during retransmission. They accomplished improved TCP performance based on CLD [7]. Initially, the network is constructed by manhattan distance, which improves the scalability. After network construction, the proposed work calculated several parameters using a fitness-based proportional fair scheduling algorithm and selected the best flow to improve the throughput and goodput. Finally, the simulation result demonstrates that the proposed work outperforms high TCP performance. An approach for improving the fairness of RTT based on the BBR CC algorithm was proposed in [8]. An enhanced BBR gamma correction algorithm is introduced for constructing a network path model. The simulation findings indicate that the fairness of RTT is enhanced when compared to the original BBR algorithm across different buffer sizes and reduces congestion. This work pays less attention to BBR was used for CC. However, it leads to an imbalance in the sending rate and transmission delay. In [9], the author introduced fast congestion handling TCP for a developed modern environment. This work developed a fast launch with agile congestion handling novel TCP design for raising the performance of short to medium TCP flows. The simulation results demonstrate that FLASH TCP achieved higher long-term and short-term bandwidth efficiency. This work pays less attention to proper user and base station management and entities are placed randomly which can cause high computational complexities and reduce the performance of TCP. In [10], scalable video coding-based streaming to overcome the head of line blocking and an intelligent stabilized. This work pays less attention to retransmission after obtaining three duplicate ACKs to reduce packet delivery failure. The multi-generative CC mechanism was proposed to be mentioned in any multi-mode GAIMD [11]. In [12], the author proposed a EWT-IoT for improving the performance of TCP. This approach proved to be more effective in CC than the others in terms of transfer time, goodput, fairness, and no losses.

3. Problem statement

The enhancement of TCP performance is influenced by various factors [13, 14]. Although the focus of existing research is on these issues, it still does not offer a suitable solution for enhancing TCP performance. However, some of the most significant issues are still unresolved by existing research work are:

In [15], an approach for improving the fairness of RTT based on the BBR CC algorithm was proposed.

- The authors only focused on the queue status and delivery rate as the primary factors for adjusting

CWND to implement CC. However, the parameter of bottleneck bandwidth and RTO was not taken into account to provide efficient CC, resulting in higher jitter and lower goodput.

- Here, BBR was used for CC. For congestion notification, criteria were not maintained and packets were randomly judged which led to an imbalance in the sending rate and transmission delay.

To address the above issue, the proposed work provides the following solutions:

- RTO is considered adaptively with the current status of the network by the proposed algorithm to improve the fairness for long-term efficient congestion control with low packet loss.
- Numerous parameters are considered in this research work for performing congestion control and avoidance such as RTT, RTO, transmission rate, bandwidth, and queue status to improve goodput.
- In the proposed work, BBR-NEnh is used for congestion control. For congestion notification, two criteria are maintained and packets are selectively judged which improves congestion control and reduces the transmission delay.

The authors introduced an adaptive congestion window method (ACW) designed specifically for the IoT environment [16].

- In this context, the CWND was adjusted in accordance with the transmission rate and available bandwidth. However, the existing CC mechanism was found to be insufficient due to the lack of crucial factors such as RTO, queue status, and RTT. Consequently, this inadequacy results in suboptimal congestion management.
- In this article, IoT devices are improperly managed and the entities are placed in a random manner that results to a significant packet loss ratio, improper CC, and computational complexity.

The proposed work provides the following solutions for the above-mentioned issues:

- Dynamic RTO is considered based on the current network status with RTT in the proposed research for effective congestion control with high throughput and goodput.
- In the proposed work, grid-based network construction is performed which reduces the complexity of congestion control and improves the performance of TCP.

In this work, authors proposed an improved fast TCP for CC in healthcare systems [17].

- In this work, missing packets, negative acknowledged packets, and duplicate packets remain in the network that leads to unnecessary traffic in TCP.
- Here, congestion window adjustment was based on propagation delay and RTT. However, it was not enough for CWND adjustment, which resulted in high transmission delay and packet loss.

The proposed work provides the following solutions for the above-mentioned issues:

- In this paper, an unwanted packet is handover to the network layer, which reduces the unnecessary traffic in TCP and problems in packet reordering.
- Here, congestion window adjustment was based on RTT, RTO, bandwidth, queue status, delay, and time interval between the two ACK of packets, which reduces the transmission delay and packet loss.

4. Proposed work

The primary objective of this research is to focus on cross-layer-based CC as well as design an efficient handover mechanism for TCP in the context of 5G IoT networks. The schematic representation of the proposed work's overall architecture is illustrated in Fig. 1. This figure describes three sequential processes including grid-based network construction, improved two-factor clustering, and Hybrid TCP congestion control.

4.1 System model

Here, the CLD is used which allows the interaction between the data link layer and the transport layer. However, it maintains layer functionality, efficient interaction among layers, and joint optimization of different layers. The major entities involved in the proposed work are IoT devices, base stations, and edge servers.

(i) IoT devices- IoT devices are the users who use a 5G connection to deliver information to objects, gadgets, and people. For the handover, user location and communication are identified here.

(ii) Base station- The base station connection between the environment's server and user sources. It allows for the low-power transmission of data across great distances.

(iii) Edge server- - It serves as a network entry point and fortifies data security and privacy protection. Clustering and secure handover are executed by an edge server, which reduces energy consumption and provides high-speed handover.

4.2 Grid-based network construction

A large number of IoT devices are arbitrarily deployed in 5G IoT, resulting in network congestion. Implement a grid-based network in a 5G IoT environment to circumvent this. An N-number of IoT devices and access nodes comprise each grid, which reduces the complexity of networks and enhances scalability. Fig. 2 represents the grid-based network

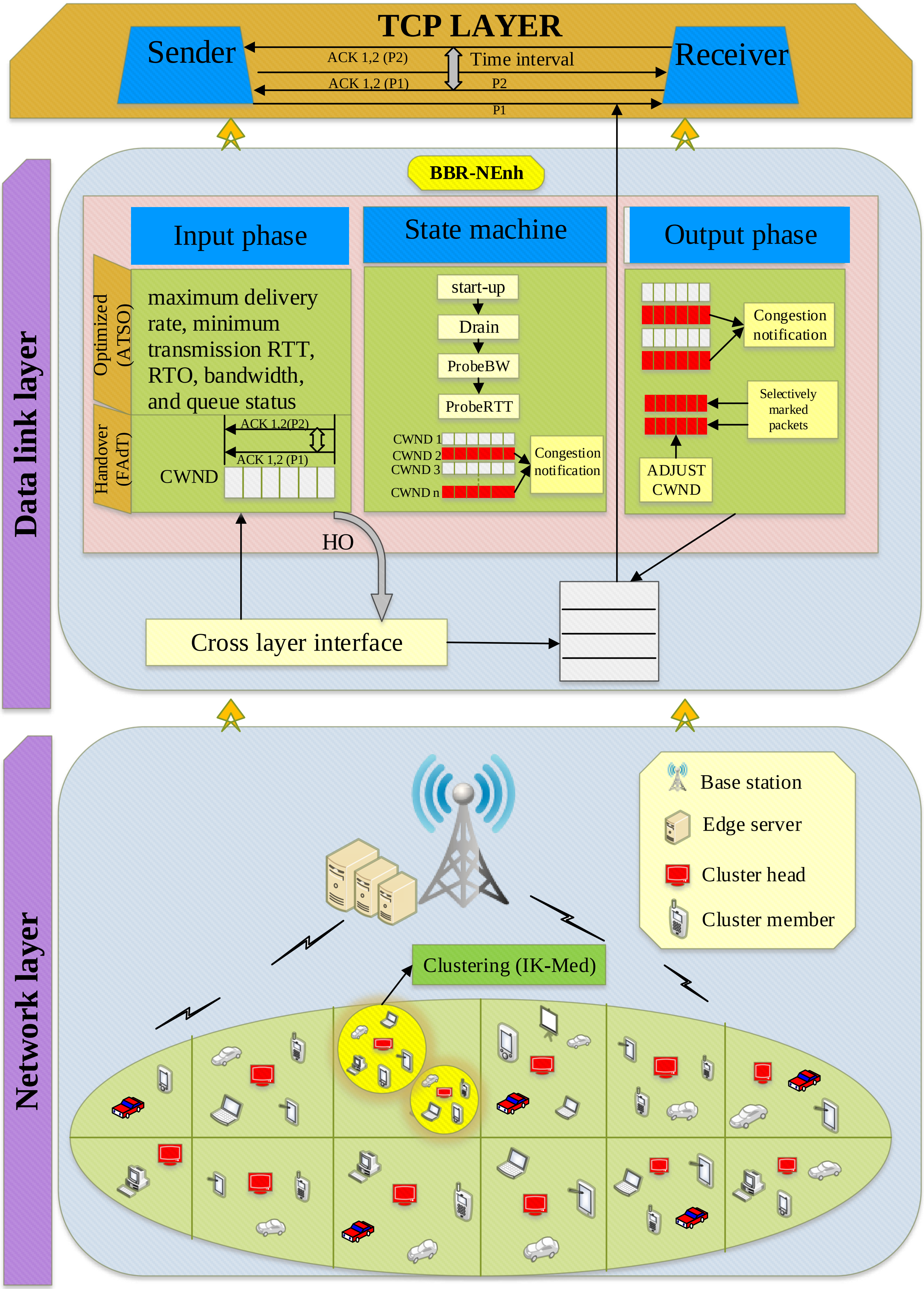

Figure. 1 Architecture of Proposed Work

Received: March 25, 2024. Revised: May 10, 2024.

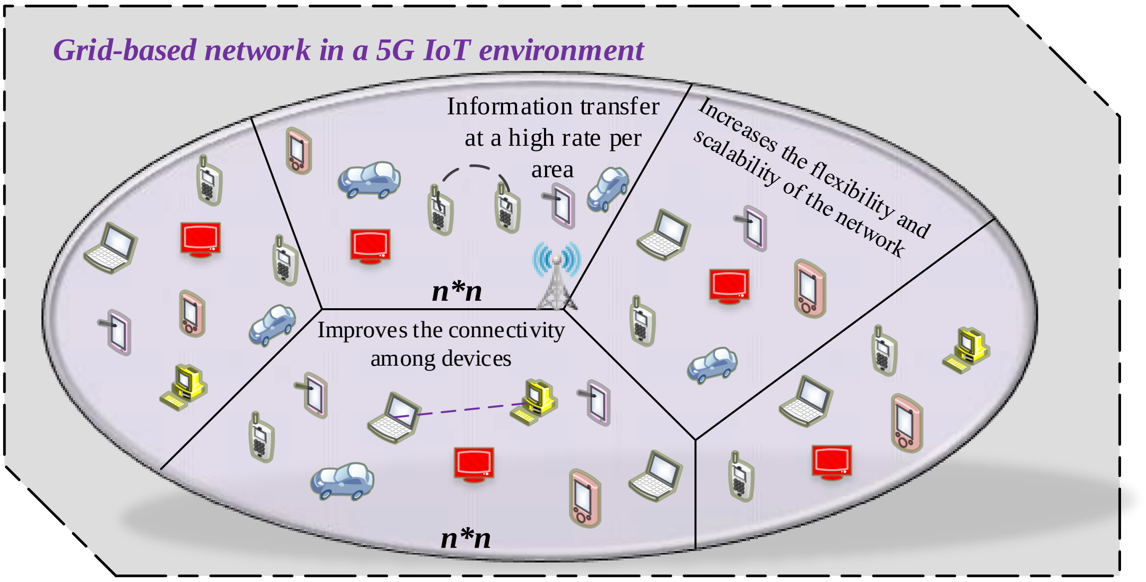

Figure. 2 Grid-based Network Construction

construction. Grid cells are partitioned into equal-sized grids, each with a unique identifier known as a grid identifier. The unique identifier of each grid cell is determined by its transmission range.

In a grid-based network topology with an n*n grid, each node can only communicate with its immediate neighbors. The computation of the grid size involves the utilization of the node location, whereas the origin position of the grid is defined as (G_0, T_0).

The grid size is denoted by the symbol ϖ which is chosen in accordance with the transmission range $F\left(\varpi = \frac{F}{\sqrt[2]{2}}\right)$. If the node is deployed, a unique grid identifier can be derived, which is determined by a calculation process.

$$GID(j,o) = \left\{(j,o) | o = \left[\frac{B-B_o}{\varpi}\right], T = \left[\frac{T-T_o}{\varpi}\right], (B_o, T_o) \epsilon \ origin\ (0,0) (B_o \leq B) \wedge (T_o T); \ \varpi : grid \text{ size} \right. \tag{1}$$

The link connectivity between devices and the users to the neighbor's users is derived from Eq. (1). If a node is located at any corner of the grid, it is capable of establishing communication with its neighboring node.

4.3 Improved two-factor clustering

The edge server implements clustering for IoT devices utilizing the IK-Med algorithm after network construction. By performing clustering techniques, energy efficiency may be enhanced, network connectivity can be improved, topology management can be made more efficient, and delay can be decreased. The proposed work clusters of IoT devices are based on two factors such as congestion score (i.e., speed limits, number of packets sent) and buffer space (i.e., queue packets in the network). After the clustering of devices, the cluster head is prioritized based on the transmission speed of data to reach its destination, distance of the user from the centroid of the cluster, processing energy, and number of devices in the cluster. In this research, an improved k-medoid clustering technique is employed to enhance the performance of clustering. This algorithm comprises two distinct steps. Initially, the time series data is partitioned into distinct subgroups according to their respective characteristics. Subsequently, the traditional k-medoids approach is employed to cluster each subgroup. Observation clustering is a technique employed to divide the given time series into discrete subgroups by leveraging the conspicuous variations in their shapes. The initial step involves the user's intuitive grouping of the provided profiles into clusters. This stage enhances the classic k-medoids clustering approach by maximizing the group of nature to determine and cluster time series data for classification and attention.

Numerous clustering algorithms are in existence, one of which is the k-medoid algorithm. This algorithm is utilized for clustering purposes, in comparison to other clustering algorithms. The determination of the appropriate number of clusters denoted as N, plays a crucial role in the k-medoids clustering process since it significantly impacts the resulting clustering outcome. Hence, the DB index is crucial in determining the optimal number of clusters.

The formal definition of the DB index can be found in Eq. (2).

$$DB = \frac{1}{H}\sum_{i=1}^{H}\max_{j\neq 1}\left(\frac{L_i+L_j}{N_{ij}}\right) \quad (2)$$

where L_i and L_j (Clustered nodes for transmission) characterize the correspondence between the phase series in the ith and jth clusters. The variation among the medoids in the ith and jth clusters is denoted by N_{ij}. Their comprehensive meanings can be spoken in subsequent reckonings Eqs. (3) and (4).

$$L_i = \frac{1}{R_i}\sum_{j=1}^{R_i} DB\left(\overrightarrow{y_j}, \overrightarrow{D_i}\right) \quad (3)$$

$$N_{ij} = DB\left(\overrightarrow{D_i}, \overrightarrow{D_j}\right) \quad (4)$$

where $\overrightarrow{y_j}$ denotes the jth time series within the ith cluster. R_i is the total number of time sequences in the ith collection. $\overrightarrow{D_i}$ represents the medoids of the ith gathering. $\overrightarrow{D_j}$ is assigned as the medoid of the jth cluster. The DB key is utilized to quantify the similarity ratio among time series inside a given set, relative to the dissimilarity within distinct clusters.

Pseudocode for clustering

```
Input: constructed node
Output: clustered node
Begin
    Initialize judge rand ()>node
    Inter cluster molecular synthesis
    While the any $N_{ij}$ change location do
        for $i \in L_i + L_j$ do (3)
            class ($Y_i$)← argmin$_j \left|Y_i - L_i + L_j\right|$
            satisfied clustering and selection
        end for
        for j ∈ $L_i + L_j$ do (4)
            $Y_j$ ← argmin$_{ij} \left|Y_i - L_i + L_j\right| - \left|Y_j - L_j + L_i\right|$
            satisfied clustering and selection
        if ($satisfied \leq not\ satisified$ ) return
        end if
        end for
    end while
end
```

The lesser the index worth is, the healthier the gathering results. The process to control the best gathering selection is hooked on differential equations and the organization and the physical and useful of the gathering with effective metrics. Calculate the DB index of each partition using k standards in the traditional k-medoids gathering procedure. Finally, the best cluster quantities are designated rendering to the DB directory value.

4.4 Hybrid TCP CC mechanism

To enhance the efficiency of TCP, the proposed work adopts a novel CC mechanism referred to as BBR-NEnh that provides high convergence and improves fairness among flows of different RTTs. Further, it includes three sub-processes such as input phase, state machine phase, and output phase. In order to determine the ideal operating point, it alternatively monitors the highest delivery rate and the shortest transmission delay. BBR regulates congestion by capping the pace at which packets are sent, limiting inflight to one BDP, as in Eq. (5).

$$BDP = Btlbw \times RTprop \quad (5)$$

BBR modifies output packet speeds to achieve the lowest predicted delivery rate. BBR maintains an active CWND to guarantee consistent throughput in delayed or aggregated ACK networks. BBR modifies CWND and pacing rate via the scaling factors cwnd_gain and pacing gain, which set up the maximum bandwidth of the previous ten round trips at Btlbw and utilize as RTprop the shortest latency observed during the preceding ten seconds.

The fundamentals of the original are optimized in the BBRV2, and the ProbeBW phase is divided into four parts. Fig. 3 represents the hybrid CC mechanism based on BBR-NEnh. Pacing rate and CWND will be matched if the bandwidth of three consecutive RTTs fails to increase by a minimum of the BBR entering the train's packet loss delivery rate and determinants of the structural and functional to the packet loss rate(PLR) or ECN mark rate. Inflight is configured to be an estimate of the maximum inflight if either the PLR or the ECN mark rate surpasses their corresponding limits. When in-flight is set or the flow exits the starting phase, the efficient continuous bandwidth detectors calculate the ECN mark rate (∂_{ecn}) using Eq. (6).

$$\partial_{ecn} = (1 - F) \times \partial_{ecn} + F \times g \quad (6)$$

where g is the weight factor and F represents the percentage of packets that are tagged in the final data window. The period of each pacing gain was divided into four cycles, and the bandwidth detection time in adaptive increases the fairness for coexisting with Reno and CUBIC in the ProbeBW phase. In order to be ready for the Up phase, the flow searches for more bandwidth and in-flight capacity during the Refill

Received: March 25, 2024. Revised: May 10, 2024.

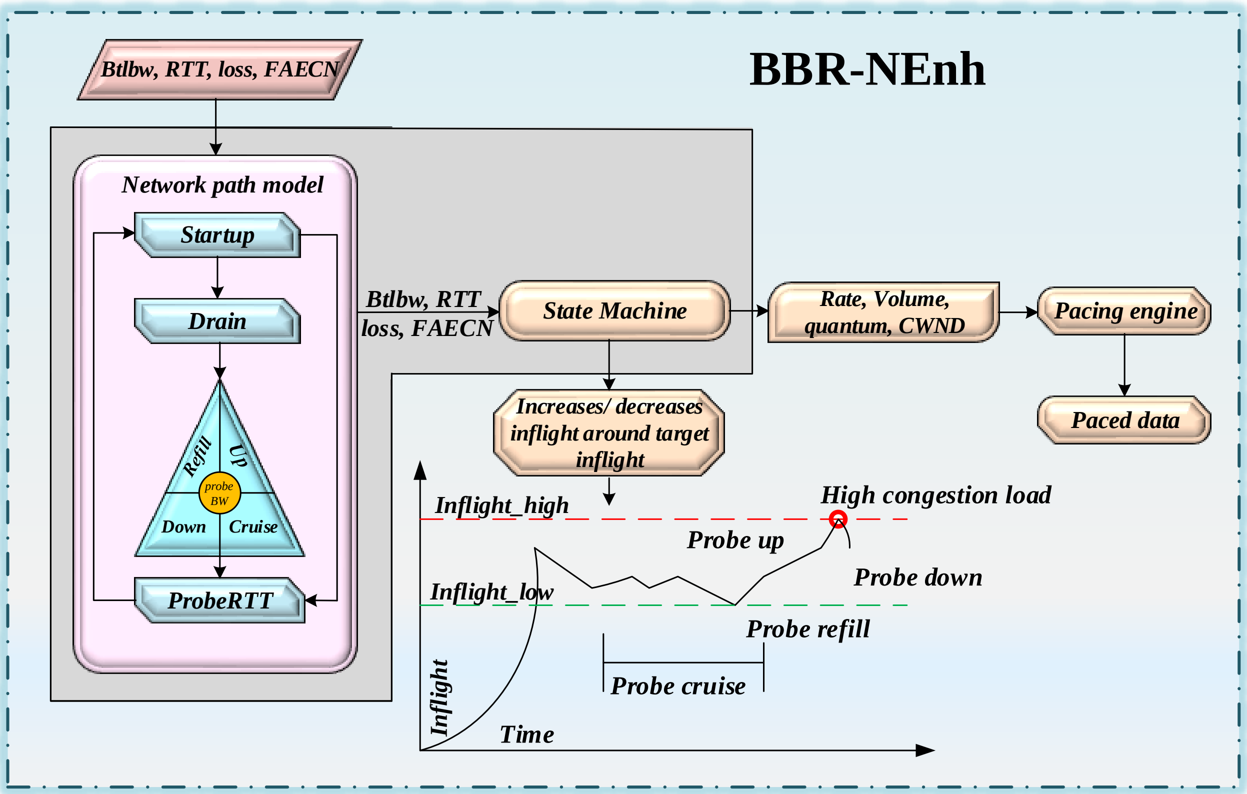

Figure. 3 Hybrid congestion control using BBR-NEnh

phase. If inflight_hi is fully utilized during the Up phase, the number of packets sent with the same RTT will increase exponentially as it encounters bottleneck connections with large buffers at various times, thereby compromising fairness. When the defined down phase is later than the current estimation of the various starting processes, the freshly established queue is cleared and the unused headroom is left. Because the ECN marking rate is present, CWND is set to the minimum between inflight, restricts the growth of packets, and leaves the phase early to prevent congestion. As a result, the increased bandwidth cannot be measured by the second start flow, which always stays at the level of the original probe. The change in RTT and queue size of the bottleneck, as well as the original ECN method's ability to provide feedback on CC for queue size and manage the mechanism for CC link in the maximum delay, is used to quantify the amount of congestion. The load factor of the bottleneck link(ξ) is calculated in Eq. (7) by dividing the current delay of "flow" over a period of five seconds by the maximum delay in the connection. This metric measures congestion as indicated by the variation in RTT.

$$\xi_i = \frac{\xi_i}{\xi_{max}} \, (\xi_i \epsilon (0,1]) \tag{7}$$

The switch reads the sending rate and evaluates which packets are a part of the high-speed flow based on the average rate's size. The switch labels the incoming packets with ECN when the queue length surpasses the specified minimum threshold N_{min} and the rate value included in the packet header is higher than the moving average. The switch modifies the mean Ave Q using packet construction and an approximation of the average sending rates of all the flows in the queue whenever a packet enters the queue, as shown in Eq. (8),

$$\text{Ave Q} = \lambda \times \delta + (1 - \lambda) \times \delta \tag{8}$$

The range of values is 0 to 1, with Q denoting the transmitting rate conveyed in the packet and denoting the weight used to update the value of Ave Q. The value in this essay is 1/8. The marking chance Y varies linearly with the queue length when the queue length shifts from U_{min} to U_{max}. The switches compare changes in the length of the current queue. ECN is not marked if the instantaneous queue length p is less than U_{min}. When ECN marks are used, they are applied in conjunction with the message's sending rate if the queue length p is more than U_min and less than U_{max}. Initial congestion in excess of the threshold is ascertained using the packet loss rate and ECN marking rate. The ECN marking threshold assigns the blind judgement for the packet threshold value for CC. High-speed flows are more likely to be marked and slowed down when congestion is an issue. The load factor divides the link's congestion level into low load and full load states.

Input phase: measurement from network traffic:- Initially, the input phase performs optimization of CC parameters such as the maximum delivery rate, RTT, minimum transmission delay, RTO, bandwidth, and queue status using ATSO [18]. The idea behind the implementation of this algorithm is to efficiently resolve intricate problems, and its characteristics facilitate adaptability, robustness, and scalability. The ATSO's exploration phase is mathematically represented as follows using Eq. (9):

$$\vec{Y}_O = \vec{Y}_O^{min} + rand \times \left(\vec{Y}_O^{max} - \vec{Y}_O^{min}\right) \qquad (9)$$

where, $\vec{Y}_O^{min}$ is the location of each tunicate and *rand* is a random quantity inside the range [0,1]. $\vec{Y}_O^{min}$ and $\vec{Y}_O^{max}$ are design variables with lower and higher bounds, individually. The tunicate adjusts its settings during the restatements using Eq. (10):

$$\vec{Y}_O(\vec{y}+1) = \frac{\vec{Y}_O(y)+\vec{Y}_O(\vec{y})}{2+b_1} \qquad (10)$$

where, b_1 is a chance amount within the range [0,1] and $\vec{Y}_O(y)$ refers to the updated position of the tunicate with respect to the location of the best topographies for seeing the best topographies for the organization represented in Eq. (11).

$$\vec{Y}_O(y) = \begin{cases} DG + B \times \left|DG - rand \times \vec{Y}_O\right| \ , if\ rand\ \geq 0.5 \\ DG - B \times \left|DG - rand \times \vec{Y}_O\right| , if\ rand < 0.5 \end{cases} \qquad (11)$$

where DG is the optimum mobbing control parameter for the organization and attention-based control mechanism for precise mobbing and the willpower for the subsequent process to prevent tunicates from hitting one another which is modelled as in Eq. (12):

$$\vec{Y}_O(\vec{y}+1) = \vec{Y}_O(g) - rand_1 \times \left|\vec{Y}_O(g) - 2 \times rand_2 \times \vec{Y}_O(\vec{y})\right| \qquad (12)$$

where $\vec{Y}_O(g)$ is a randomly chosen component from the CC in place at the time. The random numbers $rand_1$ and $rand_2$ range from 0 to 1. In addition, in the proposed ATSO, the poorest tunicate with the highest objective function value would be substituted with a randomly produced tunicate throughout each iteration. The subsequent stage of the ATSO algorithm involves adjusting their position based on the location of the best tunicate determined thus far. After parameter optimization, the Fast Adaptation Technique (FAdT) is used to dynamically execute handover, which resolves the issues with packet reordering and erroneous retransmission timeout. For a split second, FAdT delivers a sequence of two packets, each of which has been predicted to have a congestion window between ACK1 and ACK2. The packet moves on to the subsequent stages if two packets must preserve the same gap between two acknowledgements; otherwise, the network layer receives it. The TCP rapid adaption mechanism deals with packet reordering, fake RTOs, and BDP changes in a methodical manner.

Pseudocode for hybrid congestion control

```
Input: queue length of the load
Output: congestion prediction
Begin
   Initialize the RTT-us, s// update each ACK in BBR
   $U_{max} \leftarrow 0, U_{min} \leftarrow 0XFTHGVFF$
     For every ACK do
         if BBR in the start-up phase then
                RTT-us ←(now sending time)
         if RTT-us ← $max$   then
                $U_{min}$ ← RTT-us
         end if
         if RTT-us> $U_{max}$ then
                $U_{max}$ ← RTT-us
         end if
   $\emptyset$ = RTT-us/$U_{max}$ calculate the load factor
         End if
     End for
 End
```

Step 1: After the handover is identified, spurious RTO resolves the issue by delivering two identical acknowledgements to sender privileges. One acknowledgement (ACK_0) is transmitted along the old path, while the other acknowledgement (ACK_0') is sent via the new way.

Step 2: Since the dupACK (ACK_0) will always reach the sender prior to the expiration of the RTO, utilize it to reset the RTO timer to its original value to avoid data packets from naturally determined sample collected features and edges for sample allocated packets from being estimated and acknowledged.

Step 3: Data packets from the old way are returned through new packets, and the RTT sample obtained from them is RTT_0= $((RTT_{old} + RTT_{new}))^{2}$and includes smooth-end RTT records.

Step 4: $RTT_0 + RTT_{old}$ and utilise it to compute the vertical handover into the differential sample section, which can then be stated in the differential structural and featured procedures. It may also be used to reset the RTT record.

Step 5: Handover may be accompanied by a BDP change and, as previously mentioned, in the band of the structural and functional properties that may be measured by the network layer of the routing for better modifications of acknowledgment.

Step 6: Packet reordering characteristics that can be guaranteed utilising new route acknowledgement, packet delivery ratio, the transmission of the ensuing packets, and determining the resumes in accordance with a TCP receiver's typical behaviours for the ACK retaining mechanism.

Step 7: Depending on the nature of all of the states and the state-of-the-art that should be indicated in the sophisticated old TCP state to be passed over to the fresh protocol and data structures, TCP may be in one or more stages of the behavioural handover.

State machine phase- Then, in state machine phase consists of four processes such as start-up, Drain, ProbeBW, and ProbeRTT which provide queue information, explicit congestion notification, and feedback on the accurate congestion degree. By providing a congestion notification, mark it as a congestion signal to adjust the CWND and pacing rate, which reduces the packet loss rate.

Output phase: control parameters- After congestion notification, the proposed work analyzes congested streams among the streams in the proposed environment. For the data parameters of the congested stream to avoid the window constraints. Once the window size has been adjusted, a pacing process is executed to quickly provide the transmission of the data. In this way, congestion is predicted and notified. By integrating these three phases, congestion can be avoided, thereby significantly enhancing the performance of TCP.

5. Simulation results

The results of this simulation study are organized into three distinct subsections, including the simulation settings, performance analysis, and comparative analysis.

5.1 Simulation setting

Table 2. System Parameters

Hardware configuration	Hard disk	64 GB
	RAM	4 GB
Software configuration	Network simulator	NS-3.26
	OS	Ubuntu 14

Table 3. Simulation Parameter used

Parameters	**Value**
5G base station	1
#. of IoT devices	100
Mobility type	Random waypoint
Edge server	1
No. of retransmission	9
Simulation zone	$1200m \times 1600m$
Frequency bandwidth	150 MHz
Size of packets	1500 bytes
Initial energy	160J
Node mobility	7 m/s
Sachet data rate	170 Mbps
Frequency bandwidth	150 MHz

The network simulator used to execute the simulation results of the proposed research is NS-3.26. Tables 2 and 3 provide an overview of the system configuration and network parameters.

5.2 Performance analysis

The proposed research (BBR-NEnh) demonstrated improved performance across various metrics, including throughput, delay, packet loss, transmission rate, queue length, and goodput with respect to IMPRTT (15) and CW-IoT (16).

5.2.1 Impression of throughput

Throughput (ϑ^F) is described as the number of packets or requests by the user to the time occupied for data transmission that can be stated in Eq. (13),

$$\vartheta^F = \frac{packet\ size}{time\ taken} \tag{13}$$

The throughput and packet size of both the proposed and existing works are compared in Fig. 4. The results of the comparison show that the suggested work has a high throughput as compared to IMPRTT and CW-IoT of existing work. The reason for adopting BBR-NEnh for clusters of IoT devices that are based on two factors, such as congestion score and buffer space. The cluster head is prioritized based on the data transmission speed to reach its destination and the distance of the user from the cluster's centroid, processing energy, and the number of devices in the cluster. This method's metrics for packet transmission and congestion management were lower than those used in the previous clustering, which was used for CC.

The throughput comparison results show that the throughput of the proposed work for 1500 bytes of packet size is 6.5, while the existing work like IMPRTT is 5 and CW-IoT is 4.5.

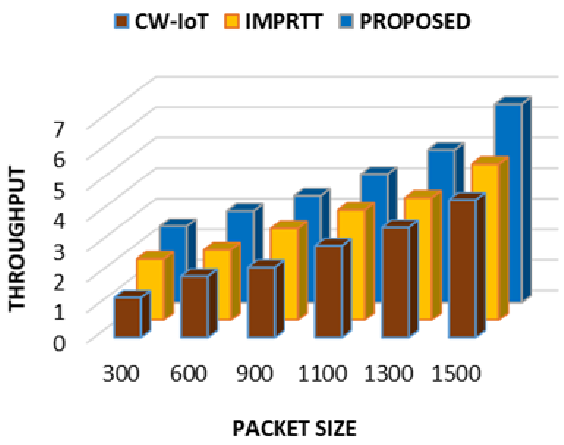

Figure. 4 Throughput Vs packet size

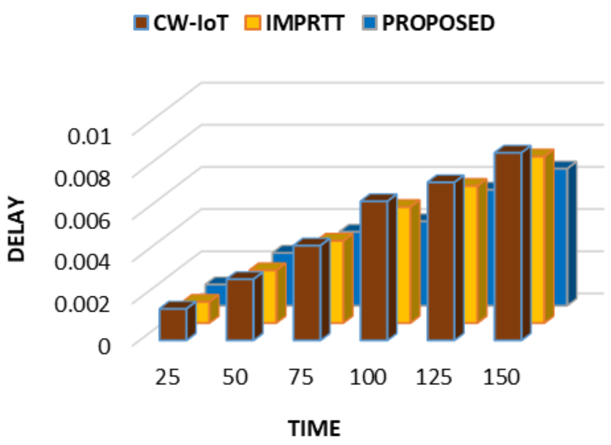

Figure. 5 Delay Vs time

5.2.2 Impression of delay

The upload and download times of the planned work's packet request and transmission are shown by the delay. To enhance the data rate, the delay time must be reduced to the absolute minimum. The delay with time of both proposed and existing works is compared in Fig. 5.

The comparison's outcome shows that the suggested method has decreased the time needed as compared to IMPRTT and CW-IoT. This study demonstrates the justification for adopting BBR-NEnh for a novel congestion management mechanism that offers high convergence and enhances fairness across flows of various RTTs as compared to existing work, consequently preventing congestion in the network. To avoid window limits for the crowded stream's data parameters. After adjusting the window size, execute a pacing process that sends the data quickly.

The proposed work's average delay over 150 seconds is 0.004, demonstrating that the proposed work outperforms earlier work like IMPRTT, which has a delay of 0.005, and CW-IoT, which has a delay of 0.006.

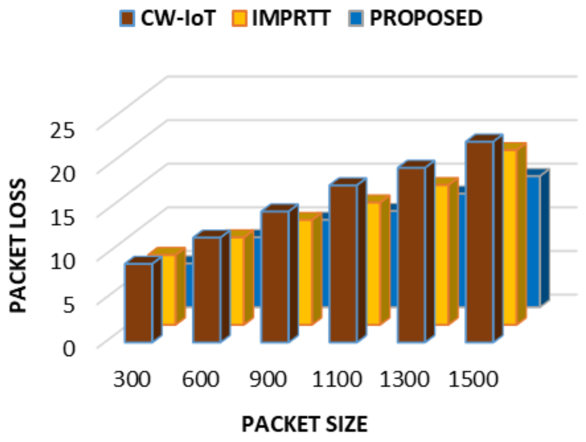

Figure. 6 Packet loss Vs packet size

5.2.3 Impression of packet loss

The amount of packet loss against the total number of packets while transmission is the average packet loss ratio. The comparison of packet loss with packet size for both planned and existing works are shown in Fig. 6. The comparison's outcome shows that the suggested work has decreased packet loss when compared to the previous work's IMPRTT and CW-IoT. This study demonstrates the benefits of adopting BBR-NEnh, which fixes the packet reordering and spurious retransmission timeout issues and is advocated by handover based on FAdT. FAdT delivers a sequence of two packets for a brief period of time. Each packet receives two acknowledgements, and the congestion window between them is determined. The packet moves on to the subsequent stages if two packets must preserve the same gap between two acknowledgements; otherwise, the network layer receives it and reduce the delay. The existing were performed retransmission of packets but the limited amount of transmission range.

The existing works, IMPRTT and CW-IoT, perform at 8 and 9, respectively, whereas the suggested BBR-NEnh solution accomplishes packet size reduction with packet loss abridged at 5. The suggested work's packet loss for 1500 bytes of packet size is 15, demonstrating that the proposed work outperforms earlier work like IMPRTT 20 and CW-IoT 23.

5.2.4 Impression of transmission rate

The packet transmission rate is the amount of data that may be transmitted across a data interface in a specific amount of time. The transmission rate of both proposed and existing infrastructure is

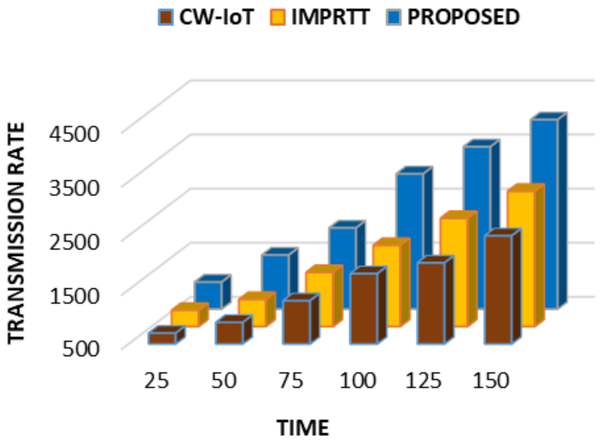

Figure. 7 Transmission rate Vs time

compared with time in Fig. 7. The results of the comparison show that the suggested work has achieved a high transmission rate as compared to IMPRTT and CW-IoT.

The transmission rate increase over time for the proposed BBR- NEnh solution is 1000, compared to the existing works' 800 and 700 transmission rates for IMPRTT and CW-IoT, respectively. The transmission rate of the suggested work for 150 of time is 4000, demonstrating that the proposed work outperforms earlier work like IMPRTT, which is 3000, and CW-IoT, which is 2500.

5.2.5 Impression of queue length

The queue length refers to the numerical value that represents the number of units that are currently in a queue or located in a certain system. Fig. 8 represents the comparison of queue length with time of both proposed and existing works.

The reason for adopting proposed novel congestion control mechanism BBR-NEnh that provides high convergence and improves fairness among flows of different RTTs when compared to conventional BBR thereby avoiding congestion in the network. Further, it includes many sub processes to reduce the queue length. The suggested work's queue length for 150 of time is 80, demonstrating that the proposed work outperforms earlier work like IMPRTT is 120 and CW-IoT is 110.

5.2.6 Impression of goodput

Goodput measures how fast and accurately useful packets are transmitted to the network and arrive at their desired location.

Fig. 9 represents the comparison of goodput with the packet size of both proposed and existing works. The reason for adopting BBR- NEnh for the proposed network is divided into n*n grids to improve scalability and reduce the complexity of the 5G IoT environment. Each grid consists of N number of IoT devices and access points. This network structure helps to improve the information transfer at a high rate per area, increases connectivity among devices, increases the flexibility and scalability of the network achieves goodput for the highly accurate determinant for the congestion control mechanism for achieving the congestion control and for the hybrid chain security purpose for the adopting congestion control.

The existing works perform the congestion control with less accurate congestion control for the signal feedback and the determination of the process less allocating the packet from the request. The existing works conduct IMPRTT is 0.3 and CW-IoT is 0.2, but the suggested BBR- NEnh method performs goodput increasing with packet size is abbreviated as 0.4. The suggested work's goodput for 1500 bytes of packet size is 1.4, demonstrating that we outperform earlier work like IMPRTT 1 and CW-IoT 0.7.

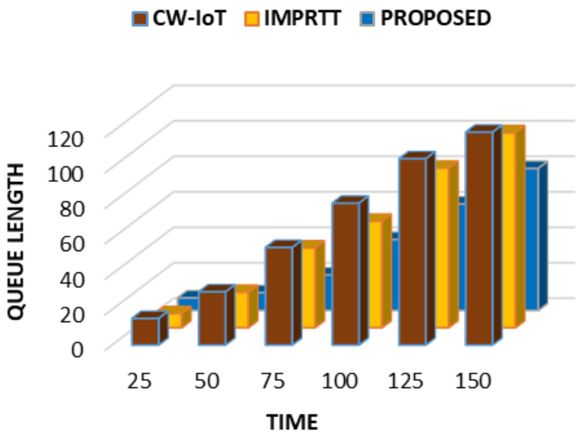

Figure. 8 Queue length Vs time

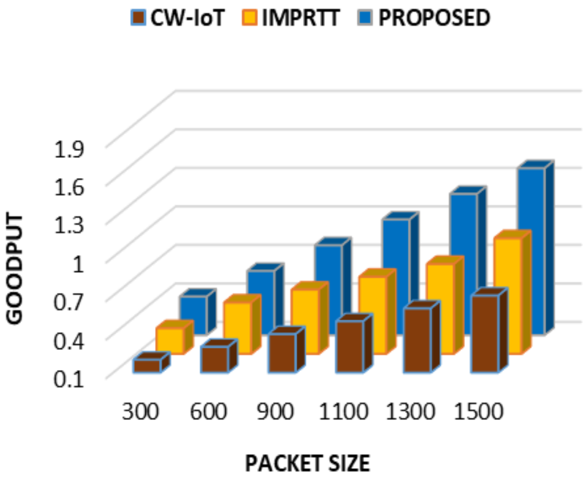

Figure. 9 Goodput Vs packet size

Table 4. Comparative Analysis

Comparison metrics	CW-IoT	IMPRTT	PROPOSED
Throughput	≃ 2.8	≃ 3.3	≃4
Delay	≃0.006	≃0.005	≃0.004
Packet loss	≃16.1	≃13.3	≃10.3
Transmission rate	≃1533	≃1800	≃ 2500
Queue length	≃67.5	≃55.5	≃36.1
Goodput	≃0.45	≃0.65	≃0.9

5.3 Comparative analysis

The comparison results indicate that the proposed BBR-NEnh framework has better performance. Table 4 shows the performance metrics in a comparative analysis of proposed and existing works.

6. Conclusion

The major challenges in TCP are congestion control and handover is addressed by proposing an efficient congestion control mechanism in TCP. Initially, grid-based network construction is performed in the network to improve scalability and reduce the complexity of the network. After network construction, two-factor clustering is performed in the network using IK-Med by considering effective metrics. The network experiences congestion as a result of the continuous transmission of packets over time, leading to a decline in performance in terms of throughput and packet delivery rate in TCP. In order to enhance the performance of TCP, a novel congestion management mechanism called BBR-NEnh is employed. This mechanism optimizes congestion parameters using ATSO and afterward performs dynamic handover based on FAdT. Moreover, it effectively addresses the issue of packet reordering and the occurrence of spurious retransmission timeouts. The experimentation of the proposed approach is executed using NS-3.26 and the evaluation of the proposed research work is carried out by comparing it with the existing approaches in terms of throughput, delay, packet loss, transmission rate, queue length, and goodput. The performance of our approach is discussed with performance analysis. The average Throughput, goodput, and packet loss of the proposed work for 1500 packets size are 4, 0.9, and 10.3, and average delay, transmission rate, and queue length over 150 seconds are 0.004, 2500, and 36.1, demonstrating that the proposed work outperforms earlier work like IMPRTT and CW-IoT. The proposed work represents a significant contribution to the field of IoT systems, offering novel solutions to improve network performance, and reliability, enhance user experience, and enable seamless connectivity in diverse environments.

Conflicts of Interest

The author declares no conflict of interest.

Author Contributions

The paper conceptualization, methodology, software, validation, formal analysis, resources, data curation, writing—original draft preparation, writing—review and editing, and visualization, have been done by 1st author.

References

[1] S. K. Bisoy, P. K. Pattnaik, M. Sain, and D. U. Jeong, "A Self-Tuning Congestion Tracking Control for TCP/AQM Network for Single and Multiple Bottleneck Topology", *IEEE Access*, Vol. 9, pp. 27723-27735, 2022.

[2] F. Chiariotti, A. Zanella, S. Kucera, and H. Claussen. "BBR-S: A low-latency BBR modification for fast-varying connections", *IEEE Access*, Vol. 9, pp. 76364-76378.

[3] B. Tierney, E. Dart, E. Kissel, and E. Adhikarla, "Exploring the BBRv2 congestion control algorithm for use on data transfer nodes", In: *Proc. of 2021 IEEE Workshop on Innovating the Network for Data-Intensive Science (INDIS)*, pp. 23-33, 2021.

[4] C. Pupiales, D. Laselva, and I. Demirkol, "Fast data recovery for improved mobility support in multiradio dual connectivity", *IEEE Access*, Vol. 10, pp.93674-93691, 2022.

[5] M. R. Kanagarathinam, S. Singh, I. Sandeep, H. Kim, M. K. Maheshwari, J. Hwang, A. Roy, and N. Saxena, "NexGen D-TCP: Next generation dynamic TCP congestion control algorithm", *IEEE Access*, Vol. 8, pp.164482-164496, 2020.

[6] S. Parween, and S. Z. Hussain. "Cross-Layer based TCP Performance Enhancement in IoT Networks", *Int. J. of Adv. Comp. Sci and Appl.*, pp.3830-396, 2022.

[7] S. Parween, and S. Z. Hussain. "A review on cross-layer design approach in WSN by different techniques", *Adv. Sci. Techn. Eng. Syst 5*, Vol. 5, No. 4, pp.741-754, 2020.

[8] W. Pan, X. Li, H. Tan, J. Xu, and X. Li, "Improvement of RTT Fairness Problem in BBR Congestion Control Algorithm by Gamma Correction", *Sensors*, Vol. 21, No. 12, p. 4128, 2021.

[9] L Guo, JYB Lee "TCP-FLASH - A Fast Reacting TCP for Modern Networks", *IEEE Access*, Vol. 9, pp.68861-68879, 2021.

[10] T. Ha, A. Masood, W. Na, and S. Cho, "Intelligent Multi-Path TCP Congestion Control for video streaming in Internet of Deep Space Things communication", *ICT Express*, Vol. 9, No. 5, pp.860-868, 2023.

[11] T. A. Edwan, I. W. Phillips, L. Guan, J. Crowcroft, A. A. Tahat, and B. E. Badr, "Revisiting legacy high-speed TCP congestion control variants: An optimisation-theoretic analysis of multi-mode TCP", *Simulation Modelling Practice and Theory*, Vol. 118, p.102542, 2022.

[12] M. Talau, TA. Herek, M. Fonseca, ECG Wille. "Improving TCP performance over a common IoT scenario using the Early Window Tailoring method", *Computer Networks*, Vol. 234, p. 109875, 2023.

[13] S. Z. Hussain, and S. Parween, "Analysis of TCP issues and their possible solutions in the Internet of things", *Int. Arab J. Inf. Tech.*, Vol. 20, No. 2, pp. 206-214, 2023.

[14] S. Z. Hussain and S. Parween, "Comparative Study of TCP Congestion Control Algorithm in IoT", *Int. Conf. on Adv. in Comp., Comm. Cont. and Net, IEEE*, pp.1428-1431, 2022.

[15] W Pan, H Tan, X Li, and X Li, "Improved RTT Fairness of BBR Congestion Control Algorithm Based on Adaptive Congestion Window", *Electronics*, Vol. 10, No. 5, p.615, 2021.

[16] R. Chappala, C. Anuradha, and PSRC Murthy, "Adaptive Congestion Window Algorithm for the Internet of Things Enabled Networks", *Int. J. of Advanced Comp. Sci. and Appl.*, Vol. 12, No. 2, 2021.

[17] J Ye, B Huang, and X Chen, "An Improved Algorithm to Enhance the Performance of FAST TCP Congestion Control for Personalized Healthcare Systems.", *Wireless Communications and Mobile Computing*, pp.1-9, 2021.

[18] A. Arabali, M. Khajehzadeh, S. Keawsawasvong, A. H. Mohammed, and B. Khan, "An adaptive tunicate swarm algorithm for optimization of shallow foundation", *IEEE Access*, Vol. 10, pp. 39204-39219, 2022.