SQUIRREL SEARCH PID CONTROLLER ALGORITHM BASED ACTIVE QUEUE MANAGEMENT TECHNIQUE FOR TCP COMMUNICATION NETWORKS

Keerthipati.Kumar [1] and R.A. KARTHIKA[2]

Computer Science and Engineering, VELS Institute of Science, Technology & Advanced Studies (VISTAS),
Chennai, INDIA.

Summary

Active queue management (AQM) is a leading congestion control system, which can keep smaller queuing delay, less packet loss with better network utilization and throughput by intentionally dropping the packets at the intermediate hubs in TCP/IP (transmission control protocol/Internet protocol) networks. To accelerate the responsiveness of AQM framework, proportional-integral-differential (PID) controllers are utilized. In spite of its simplicity, it can effectively take care of a range of complex problems; however it is a lot complicated to track down optimal PID parameters with conventional procedures. A few new strategies have been grown as of late to adjust the PID controller parameters. Therefore, in this paper, we have developed a Squirrel search based PID controller to dynamically find its controller gain parameters for AQM. The controller gain parameters are decided based on minimizing the integrated-absolute error (IAE) in order to ensure less packet loss, high link utilization and a stable queue length in favor of TCP networks.

Keywords:

Transmission Control Protocol (TCP); Active queue management (AQM); proportional-integral-differential (PID); integrated-absolute error (IAE); Squirrel search Algorithm (SSA); Squirrel search based PID (SS-PID).

1. Introduction

In today's Internet environment, congestion due to network traffic brings about extended time delay in information transmission and often creates the queue length within the buffer of intermediate router (switch) overflow, and be able to still prompt network breakdown due to the unpredictable interference and the rapid growth of enormous number of clients [1], [2]. To solve this, congestion control mechanisms that decide a queuing principle for routers (switches) ought to be projected by showing the order in which the packets are to be transported and to signify which set of packets ought to be ignored whenever congestion happens. The congestion control strategy called active queue management (AQM) has been proposed to manage this issue. AQM is a proficient way that identifies inceptive congestion and provides quick notification of states of data for the present Internet circumstance by dropping the packets in bottleneck router before its buffer turns out to be full to keep away from severe congestion of TCP flows. By dropping the packets before buffer overflow happens, AQM tries to keep away from global synchronization and huge queueing delay [3-5].

Much research has been dedicated to AQM algorithm [6-9]. They can be grouped into rate based, queue based, and joined rate-queue based according to their congestion recognition strategies. Random Early Detection (RED) and proportional integral (PI) are queue-based. Green, BLUE and adaptive virtual (AVQ) fits in rate-based class. The joined queue-rate-based category incorporates random exponential (REM) [10], Yellow, and RaQ. An alternative classification into time and event driven AQM plans depends on the technique for updating marking probability. In these algorithms, the most notable event-driven AQM method is RED algorithm, which has been broadly utilized in routers [11] [12]. RED attempts to keep away from the network congestion, and it has a enhanced design to queue length while the queue buffer isn't full. In this way, probability of packet dropping increments as the route queue length increments [13].

However, the parametric values used by these algorithms are more responsive to the change of network traffic due to insufficient analysis and design of these algorithms. Although, PI algorithm [14] has benefit over RED algorithm, due to the minute queue length oscillation in routers, its proportional and integral coefficients are static but its dynamic performance is hard to ensure.

As a strategy to understand a stable congestion control framework, an AQM dependent on control theory and proportional-integral-differential (PID) controller has been proposed. AQM dependent on control theory uses a parameter called target queue length, which is a value that the framework endeavors to keep the actual queue length close to. Customarily, this value has been set to half the buffering limit, consequently making an empty space in the router's buffer. However, the dynamic control of the target queue length successfully uses the buffering limit of the bottleneck router [15]. Likewise, in spite of less complication, it can effectively work out a range of complex problems. Be that as it may, it is hard to track down optimal PID parameters with conventional methods. A few new strategies such as Genetic Algorithms and ant colony optimization have

---

Manuscript received April 5, 2023
Manuscript revised April 20, 2023
https://doi.org/10.22937/IJCSNS.2023.23.4.16

been grown as of late to adjust the PID controller's parameters [16].

1.1. Problem statement and contribution

Proportional integral derivatives (PID) have benefit over PI algorithm through its dynamic performance. Be that as it may, the technique used for tuning of coefficients in PID algorithm is somewhat complicated. By improving the performance of PID algorithm, congestion in the network can be controlled. Along these lines, to develop the performance of PID algorithm, the accompanying contributions are introduced in this paper.

By tuning the PID algorithm coefficients, the performance of the PID algorithm for AQM can be enhanced. Thus, Squirrel Search algorithm (SSA) for optimizing or tuning the coefficients in PID algorithm.

The proposed algorithm is introduced for optimizing the coefficients such as proportional coefficient (G_P), integral coefficient (G_I) and derivative coefficient (G_D) in PID algorithm.

This proposed approach is carried out in the platform of NS2.

The performance of proposed approach is assessed in terms of delivery ratio, delay

2. Literature Review

Maurizio Casoni et al. [17], designed a queue management algorithm called PINK (Passive INverse feedbacK) to impose a specific resource allocation strategy indirectly on characterized sets of client hosts. PINK put in intelligence at intermediate hubs that associate client hosts either to external networks or bottleneck links frequently, permitting these hubs to adjust the TCP Acknowledgements (ACKs) going through dynamically. Moreover, to uphold a specific bandwidth utilization upper bound, the adjustment comprises in substituting advertised Receive Window fields (RCV.WNDs) with custom values; also, PINK requires just the flows RTTs, number of active connections, and the transmission channel bandwidth to process fresh RCV.WND values. Further, without the participation of clients, PINK allows to impose a centralized bandwidth management, which implies that no adjustment to end hosts was required.

Yang Liu et al. [18], proposed an H∞ congestion tracking control issue for nonlinear TCP/AQM network. An adapted TCP/AQM network model with modelling uncertainty and external disturbance was first presented by the inspiration from current network models. And afterward, a function was characterized interestingly for the preferred queue length, which knows how to simulate the progressions in the queue length during a day. Subsequently, by consolidating H∞ theory and integral backstepping procedure, an AQM system was introduced to control network congestion.

Sana Sabah Sabry et al. [19], deployed the active queue management utilizing Grey Wolf Optimizer for tuning the PID controller gains in an optimal manner so as to ensure lower packet loss rate, stable queue length and avoid network congestion. Moreover, the controller was examined and benchmarked against a traditional PI controller to show its robustness through MATLAB simulation experiments.

Sukant Kishoro Bisoy et al. [20] proposed a proportional-differential-type feedback controller called Novel-PD as AQM to manage the queue length with little oscillation. It estimates the present queue length and modifies the packet drop probability dynamically with the help of the estimated present queue length and differential error signals.

To take care of the network congestion problem occurring at multiple routers, Lujuan Ma et al. [21], proposed a congestion control algorithm. The model employed captures the TCP elements and faces the four potential practices of TCP network whenever congestion happens. The role of TCP/AQM was to diminish congestion by effectively eliminating certain packets in the queue. Considering the packets probability set apart as dropping like the control input, the congestion problem was addressed by utilizing the backstepping technique to maintain the queue length converge to the preferred queue length. Here, to deal with the AQM issue, the traditional backstepping based control methodology was reached out to TCP network with numerous bottleneck routers. Further, a modified model was presented for TCP/AQM network with different bottleneck routers, as roused from the existing network model with single bottleneck router. And afterward, to manage network congestion dependent on integral backstepping, an AQM strategy Backstepping Congestion Control for Multi-router (BCCM) was proposed.

Kun Wang et al. [22], presented a more accurate and general network congestion algorithm for TCP/AQM system. In addition, an adaptive congestion controller was planned by desirable quality of the Barrier Lyapunov Function (BLF), backstepping-like and Neural Networks (NNs) approximation procedures, where the steady and transient state performances resting on the tracking error can be pre-assigned and different signals of the closed-loop system, are additionally confirmed to be uniformly, semi-globally, and ultimately bounded.

Kun Wang et al. [23], considered a performance constraint control problem for TCP/AQM network with input saturation and external disturbance. An adaptive fuzzy controller with agreed constraint was accomplished based on backstepping-like plan strategy and fuzzy approximation method, to guarantee that the transient and steady state performances of the tracking errors be able to be fulfilled. Further, the stability analysis demonstrates that every signals in the closed-loop framework are uniformly, semi-globally, and ultimately bounded.

3. Squirrel Search PID Algorithm Based AQM Technique in TCP communication networks

Active Queue Management (AQM) can keep up elevated throughput and lesser packet queuing delay in routers is an active area of study in the previous few years. In TCP/IP networks, AQM functions as a primary congestion control plot for improving network utilization and lessening of packet loss. Here, to guarantee low packet loss, stable queue length, and high link utilization. The PID controller is utilized as an active queue manager for Internet routers. However, the PID controllers show poor control performances for the integrating and enormous time delay processes. Therefore, we have proposed an ideal PID controller based on Squirrel Search algorithm for choosing the PID controller gain parametric values. Here, the Structure of Squirrel Search algorithm based PID controller design is given in below figure 1.

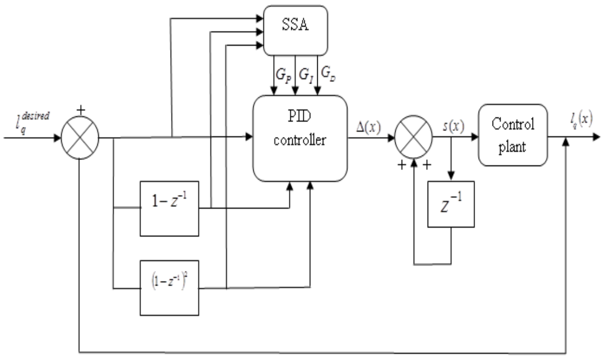

Fig. 1. Structure of Squirrel Search algorithm based PID controller design

In above figure, $l_q(x)$ represents the queue length at time x; $l_q^{desired}$ denotes the target queue length; G_P, G_I and G_D represents the controller gain parameters. Here, the controller gain parametric values are adjusted through the squirrel search algorithm so as to develop the controlling performance of the PID controller.

3.1 System Model of PID based TCP networks

To accurately reflect the dynamics of TCP, a window-based nonlinear fluid-flow dynamic model is considered for TCP networks, which have the average TCP window size 'A' and average queue length 'l_q' (both 'A' and 'l_q' are represented in packets) through coupled non-linear differential equations are specified as below:

$$\dot{A}(x)=\frac{1}{\frac{l_q(x)}{L_c}+D_{prop}}-\frac{A(x)}{2}\frac{A(x-T(x))}{\frac{l_q(x-T(x))}{L_c}+D_{prop}}prob(x-T(x)) \tag{1}$$

$$l_q(x)=\begin{cases}-L_c+\dfrac{M(x)}{\frac{l_q(x)}{L_c}+D_{prop}}A(x) & if\ l_q(x)>0\\ \max\left\{0,-L_c+\dfrac{M(x)}{\frac{l_q(x)}{L_c}+D_{prop}}A(x)\right\} & if\ l_q(x)=0\end{cases} \tag{2}$$

In the above equations (1) and (2), D_{prop} represents the propagation delay (in seconds); M denoting the number of TCP connections; $prob$ symbolizes the packet dropping probability (i.e. the parameter used in order to the control the sending rate by maintaining the bottleneck queue length). Also, T is the round trip time during transmission, which is illustrated as,

$$T=\frac{l_q}{L_c}+D_{prop} \tag{3}$$

Where, L_c denotes the link capacity (in packets/sec). Moreover, the effects of propagation delays and packet-dropping probability in TCP is made clear by modeling the TCP AQM network, the same way as a time-delayed system having a saturated input. Here, the saturated input based on packet-dropping probability having value among 0 and 1 can be expressed by,

$$s(x)=prob(x-T(x)) \tag{4}$$

To ensure system stability, the PID controller generates saturated input $s(x)$ as control input based on the output error signal $\Delta(x)$ as,

$$\Delta(x)=l_q(x)-l_q^{(desired)} \tag{5}$$

Where, $l_q^{(desired)}$ represents the target queue length

Therefore, a PID controller with its associated input $s(x)$ is adjusted by the output error signal $\Delta(x)$ is represented as:

$$s(x) = G_P\left[\Delta(x) + \frac{1}{Y_I}\int_0^x \Delta(\gamma)d\gamma + Y_D\frac{d}{dx}\Delta(x)\right]$$

(6)

Where, G_P denotes the proportional gain, Y_I and Y_D represents the integral and derivative time constants respectively. Now, eqn. (6) is rewritten as,

$$s(x) = G_P\Delta(x) + G_I\int_0^x \Delta(\gamma)d\gamma + G_D\frac{d}{dx}\Delta(x) \qquad (7)$$

Where, $G_I = \frac{G_P}{Y_I}$ and $G_D = G_P Y_D$ are the integral gain and the derivative gain.

However, the performance of the PID controller is decided by the integral absolute error (IAE) measure, which is found by the optimal representation of the controller gain values like G_P, G_I and G_D . Thus, the proposed SS-PID (Squirrel search PID) controller is designed to find the optimal controller gain parameters with the objective function ' O ' given as,

$$O := \min(IAE) = \int_0^\infty |\Delta(\gamma)|d\gamma$$

(8)

3.2 Squirrel search algorithm (SSA) for controller gain optimization

Squirrel search algorithm is developed from the strategy of foraging behavior followed by squirrels. The food search process isn't simpler even during warm climate (autumn), since the squirrels have to investigate various areas of forest and just glide from one tree to the next for looking through food resources. Anyway they get food from the plentifully available acorns. The squirrels consume acorns directly and begin looking for the optimal food source (hickory nuts) intended for to be stored for winter season. This procedure of squirrels will assist them with satisfying their energy necessity during exceptionally harsh weather and accordingly the probability of survival is expanded. As the winter seasons brings about loss of leaf cover, the danger of predation is high. In this manner, the squirrels are less active (goes to hibernate mode) and decrease their foraging trips during such situations. During the ending of winter season, they are back to ordinary and become active in order to search food for their survival. The above process is repeated all through their lifecycle and this cyclic process prompts the establishment of SSA.

In the numerical model of SSA, a portion of the assumptions are made as of:

Total count of squirrels is ' M ' and everyone is thought to be on one tree.

Food search of each squirrel is individual and applies dynamic foraging behaviour for choosing the ideal food source

Three sorts of trees like normal tree, oak tree with acorn nuts food source and hickory tree with hickory nuts food source are only available in forest.

Within search space, one hickory tree and three oak trees are available.

For instance, if the number of squirrels be, (say, M =50), their food resources are gotten from 3 acorn nut trees and 1 hickory nut tree, while the remaining 46 trees have no food source. From the entire population of squirrels, around 92% of them are on normal trees. Fittest squirrels discover optimal food sources and increase their survival rate. The algorithm is applied in our proposed PID controller algorithm to find the fittest solution (i.e. control parameters) based on the foraging behavior of squirrels. Steps involved in the proposed SS-PID algorithm is given beneath,

3.2.1 Random initialization

During initialization, M number of flying squirrels S in different locations is randomly initialized as,

$$S_M = \begin{bmatrix} S_{1,1} & S_{1,2} & \cdots & \cdots & S_{1,z} \\ S_{2,1} & S_{2,2} & \cdots & \cdots & S_{2,z} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ S_{m,1} & S_{m,2} & \cdots & \cdots & S_{m,z} \end{bmatrix} \qquad (9)$$

In above equation, each value $S_{l,k}$ represents the location of l^{th} squirrel in k^{th} dimension. The Flowchart of proposed Squirrel search based PID controller is given in the following figure 2.

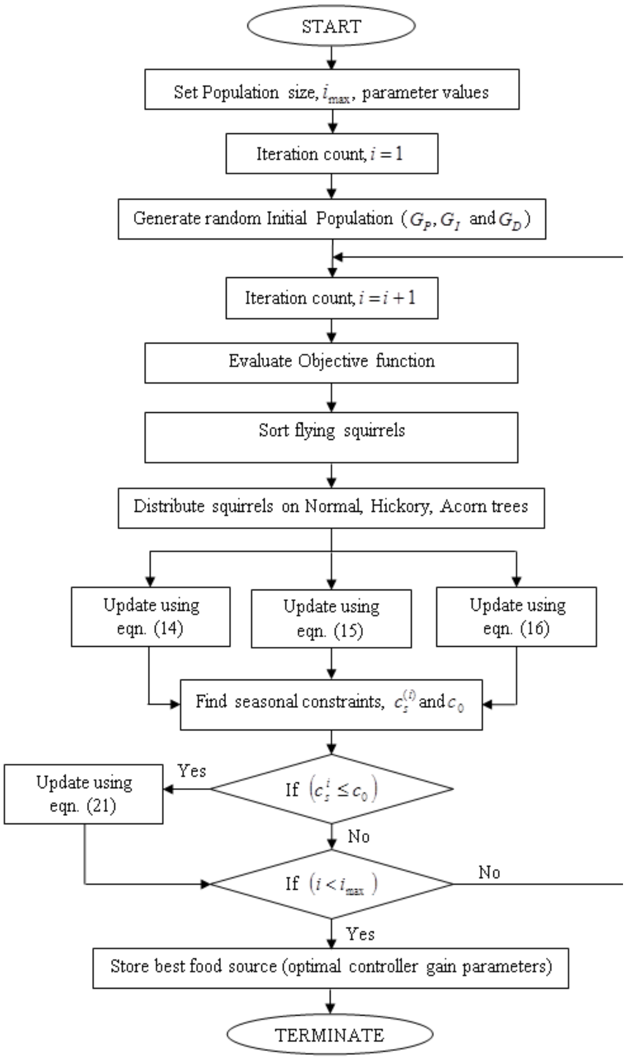

Fig. 2. Flowchart of Squirrel search based PID controller

Also, the random location of squirrels is allocated based on uniform distribution function applied within the minimum $S_{\min}$ and maximum $S_{\max}$ range of l^{th} squirrel in k^{th} dimension is given as,

$$S_l = S_{\min} + v(0,1) \times (S_{\max} - S_{\min}) \tag{10}$$

Where, $v(0,1)$ is the uniformly distributed arbitrary number within range $[0,1]$.

Moreover In the above equation (9), each row of the initial solution illustrates the controller gain parameters vector generated as,

$$S = \begin{bmatrix} G_P & G_I & G_D \end{bmatrix} \tag{11}$$

Also, each controller gain parameter having set of bits are represented as,

$$S = \begin{bmatrix} \underbrace{S_{1,1}, S_{1,2}, \ldots S_{1,x}}_{a_1=G_P} & \underbrace{S_{1,x+1}, S_{1,x+2}, \ldots S_{1,y}}_{a_2=G_I} & \underbrace{S_{1,y+1}, S_{1,y+2}, \ldots S_{1,z}}_{a_3=G_D} \end{bmatrix} \tag{12}$$

3.2.2 Fitness evaluation

During this stage, the fitness of each and every solution is evaluated. The fitness calculation is given as,

$$F = \begin{bmatrix} F\left(\left[S_{1,1}, S_{1,2}, \cdots, S_{1,z}\right]\right) \\ F\left(\left[S_{1,1}, S_{1,2}, \cdots, S_{1,z}\right]\right) \\ \vdots \\ \vdots \\ F\left(\left[S_{1,1}, S_{1,2}, \cdots, S_{1,z}\right]\right) \end{bmatrix} \tag{13}$$

In the above equation, the fitness of every solution is calculated with the help of eqn (8). By this way, the optimal controller gain parametric values like, G_P, G_I and G_D are found.

Sort, declare and random select process

Once after finding the fitness values of every solution (i.e. squirrel's location), the values are sorted in ascending order. It is considered that the fittest position (i.e. minimal IAE) is recognized as to be on hickory nut tree. Also, the subsequent three best solutions are assumed to be on acorn nuts trees and the squirrels tries to go in the direction of the hickory nut tree. Left over squirrels is assumed to be on normal trees. Once the declaration step is done, a random selection process is done to choose some squirrels to travel towards the hickory nut tree. However, all the squirrels are affected by predators and thus the algorithm is modeled based on the predator presence probability (p_{prob}).

Generate new locations

Once the assumptions about the flying squirrels is made at the declaration stage, the dynamic foraging behaviour based on the predator presence probability (p_{prob}) is mathematically formulated as follows,

Norm 1: Flying squirrels on acorn nut trees (S_A^i) tends to move towards hickory nut tree.

$$S_A^{i+1} = \begin{cases} S_A^i + g_{dist} \times g_{const} \times \left(S_H^i - S_A^i\right) & ; r_1 \ge p_{prob} \\ Rand & ; otherwise \end{cases} \tag{14}$$

Norm 2: Flying squirrels resting on normal trees (S_N^i) be likely to move towards acorn nut trees.

$$S_N^{i+1} = \begin{cases} S_N^i + g_{dist} \times g_{const} \times \left(S_A^i - S_N^i\right) & ; r_2 \geq p_{prob} \\ Rand & ; otherwise \end{cases} \tag{15}$$

Norm 3: Flying squirrels on top of normal trees (S_N^i) trying to move towards hickory nut trees.

$$S_N^{i+1} = \begin{cases} S_N^i + g_{dist} \times g_{const} \times \left(S_H^i - S_N^i\right) & ; r_3 \geq p_{prob} \\ Rand & ; otherwise \end{cases} \tag{16}$$

In the above equations (14), (15) and (16), g_{dist} represents the random gliding distance; $\left(g_{const} = 1.9\right)$ denotes the gliding constant that helps controlling between exploration and exploitation phases of squirrels; i and $i+1$ defines the current and next iteration; S_A^i, S_H^i and S_N^i represents the position of the flying squirrel that arrived acorn nut tree, hickory nut tree and normal trees respectively. Also, r_1, r_2 and r_3 are random numbers within range [0, 1].

3.2.3 Aerodynamics of gliding

Gliding mechanism is illustrated by the resultant force produced by the lift (E) and drag (d) force of the flying squirrels. Thus, the lift-to-drag ratio (also termed as, glide ratio) of the flying squirrel gliding at steady speed descending at horizontal angle is defined as,

$$G = \frac{E}{d} = \frac{1}{\tan \psi} \tag{17}$$

In above equation, ψ represents the glide angle, and can be given as,

$$\psi = \arctan\left(\frac{d}{E}\right) \tag{18}$$

Here, the lower value of ψ helps in increasing the glide path length of the squirrels. Moreover, the lift force generated from the downward deflection of air flowing through the wings is given as,

$$E = \frac{1}{2\phi E_{coef} T^2 L} \tag{19}$$

Further, the frictional drag is given as,

$$d = \frac{1}{2\phi d_{coef} T^2 L} \tag{20}$$

Where, $\phi = (1.204 kgm^{-3})$ represents the density of air; $T = 5.25 ms^{-1}$ denotes the speed; $L = 154\, cm^2$ specifies the surface area of the body; E_{coef} and d_{coef} represents the lift coefficient and drag coefficients respectively.

3.2.4 Seasonal monitoring condition

As the seasonal change plays an important role in the foraging activity of flying squirrels, it is necessary to add a seasonal monitoring condition to make the algorithm more realistic. To mathematically represent the seasonal monitoring condition, few steps are followed as,

```
  evaluate seasonal constant ( $c_s^{(i)}$ ) by,
  $c_s^{(i)} = \sqrt{\sum_{j=1}^{z} \left(S_{A,j}^i - S_{H,j}^i\right)^2}$ , where $i = 1,2,3$
represents iteration count
  Find minimum seasonal constant using,
  $c_0 = \frac{10e^{-6}}{(365)^{\frac{2.5i}{i_{\max}}}}$ , where $i_{\max}$ represents maximum
iteration
  // Analyze seasonal monitoring condition
  if $\left(c_s^i \leq c_0\right)$
  Perform random relocation
  Else $\left(c_s^i > c_0\right)$
  Flying squirrels itself explores for optimal food source
```

3.2.5 Random relocation during winter season

Based on the seasonal monitoring condition, the active squirrels that are able to explore their optimal food source are found. Moreover, the squirrels which are still survived but cannot explore are also found and made to explore their optimal food source from the random relocation strategy. Here, the random relocation is based on the levy flight distribution method, that can be formulated as,

$$S_N^* = S_{\min} + levy(m) \times \left(S_{\max} - S_{\min}\right) \tag{21}$$

Where, $levy(y)$ represents the levy distribution as:

$$levy(y) = 0.01 \times \frac{u_p \times \delta}{\left|u_q\right|^{1/\chi}} \tag{22}$$

In the above levy distribution representation, the parameter $\chi = 1.5$ is a constant and u_p and u_q denotes the random numbers distributed normally within [0,1]. Also, δ is found as,

$$\delta = \left(\frac{\Gamma(1+\chi)\times \sin\left(\pi\chi/2\right)}{\Gamma\left(\frac{1+\chi}{2}\right)\times\chi\times 2^{\left(\chi-1/2\right)}} \right)^{\frac{1}{\chi}}$$

,where:

$$\Gamma(y) = (y-1)! \qquad (23)$$

3.2.6 Stopping criterion

Stopping criterion is made when the optimal solution is found or the maximum number of iterations is attained. At the end of SSA, the optimal controller gain parameters are found, which are then applied to the PID controller so as to advance the controlling efficiency of the PID controllers.

Algorithm 1: Steps in proposed Squirrel Search algorithm based PID controller design

---

1. Predefine the search range for the values of controller gain parameters (G_P, G_I and G_D) of the controller
2. Design an objective function using Eq. (8).
3. Set the population size 'M' and make 'M' random initial solutions as like eq. (9).
4. Now find the fitness function for each solution
5. Sort every solution based on fitness
6. Store best solution (i.e. controller gain parametric values)
7. Construct new set of solutions based on predator presence probability (p_{prob})
8. Evaluate seasonal monitoring condition
9. If $\left(c_s^i \le c_0\right)$, Perform random relocation using eq. (21)
10. Again find fitness function for the newer solutions
11. Compare with previous best solution
12. Replace and store if current solution is better than the previous solution
13. Repeat
14. If termination criteria is met
15. Get final best controller gain parameters and apply in SS-PID controller
16. End

---

4. Results and Discussion

The results talked about in this section were acquired from the proposed SS-PID controller design for TCP/AQM implemented in a PC with the accompanying details: CPU Intel® Pentium 1.9 GHz, 64-bit operating system, Microsoft® Windows 10, 4 GB of RAM, and NS-2 (Network Simulator-2) platform.

For TCP/AQM network, we have considered 100 homogeneous TCP connections being contributed by one bottleneck link having capacity of 10 Mbps, i.e. $L_c = 1250\,(\text{packets/second})$ with propagation delay of $D_{prop} = 0.08$ and the target queue size, $l_q^{desired} = 150\ packets$. Moreover, the population size of SS algorithm is set as, $M = 50$.

The performance of the proposed SS-PID controller is demonstrated through a progression of numerical simulations carried by out using NS-2 with the dumbbell network topology is shown in Figure 3.

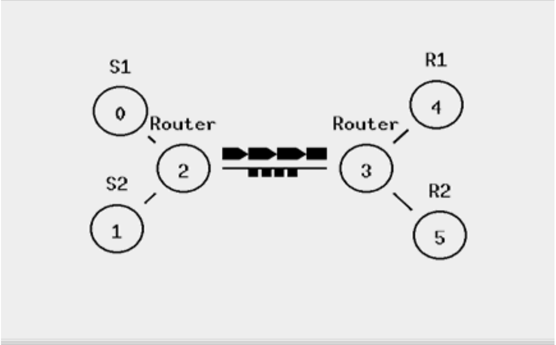

Fig. 3. Simulation topology

4.1. Performance analysis

The performance of the proposed Squirrel Search based PID (SS-PID) controller model is evaluated in terms of delay (ms), Delivery Ratio, Link utilization (%) and Packet Loss (%). Here, the delay (ms) and Delivery Ratio are analyzed by varying Time (ms). Also, the Packet Loss (%), and Link utilization (%) metrics are analyzed by varying Round Trip propagation delay (ms) the Number of TCP Connections. Besides all the performance measures calculated for the proposed SS-PID controller is compared with that of the conventional PID and PI controllers.

4.1.1 Analysis made in terms of delay (ms)

In this section, the delay occurred during packet transmission is analyzed when time progresses. Table 1 depicts the delay occurred with respect to time for the proposed and the existing controller models. Moreover, the graphical representation of the comparison made is given in figure 4. From table 1, it is clear that the delay caused by the proposed SS-PID is very less when compared to the existing algorithms like PID and PI controllers. Also, the delay is increased when the time progresses but however lower than the existing algorithms.

Table 1: Time (ms) Vs Delay (ms) for proposed and existing algorithms

Time (ms)	SS-PID	PID	PI
20	2	5	6
40	6	8	9
60	10	11	14
80	13	14	19
100	17	17	24

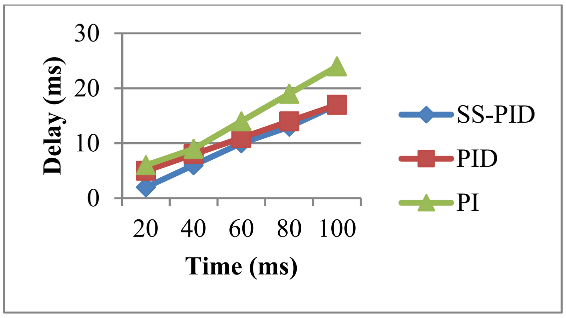

Fig. 4. Graphical representation of Time (ms) Vs Delay (ms)

4.1.2 Analysis made in terms of Delivery Ratio

Here, the performance of proposed SS-PID controller is investigated in terms of Delivery Ratio with respect to Time (ms). Similarly, the comparison is made for proposed and existing algorithms and given in table 2. Moreover, the Graphical representation of the compared values is given in figure 5. From the below table, it is noted that, the delivery ratio is improved for the proposed SS-PID controller than the conventional algorithms. The maximum delivery ratio is obtained at time=20ms, while it is less for the existing algorithms. The delivery ratio is decreased when time increases. However, the delivery ratio is better than the existing algorithms, showing better data transmission rates by the proposed SS-PID controller design.

Table 2: Time (ms) Vs Delivery Ratio for proposed and existing algorithms

Time (ms)	SS-PID	PID	PI
20	0.98	0.84	0.76
40	0.892361427	0.813673	0.697417
60	0.804281526	0.766997	0.649702
80	0.759346424	0.699434	0.603553
100	0.682954667	0.672706	0.53736

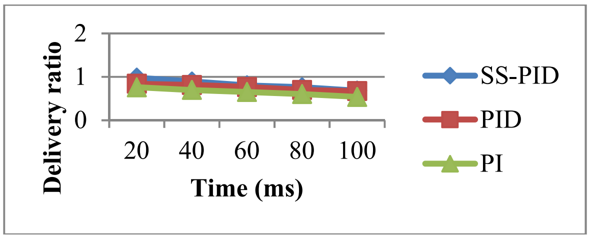

Fig. 5. Graphical representation of Time (ms) Vs Delivery Ratio

4.1.3. Analysis made in terms of Packet Loss (%) for varying Number of TCP Connections and Round Trip propagation delay (ms)

Here, the Packet Loss measure is first analyzed by modifying the quantity of TCP Connections and the values obtained are given in table 3. Moreover, the attained values are plotted in figure 6. Here, the total number of TCP connections are varied as, 50, 100, 150, 200 and 250 respectively. From table 3, it is shown that the packet loss faced by the proposed SS-PID is lower than the conventional methods. Also, the packet loss values are increased when the number of TCP connections is raised; but not higher than the existing methods.

Table 3: Number of TCP Connections Vs Packet Loss (%) for proposed and existing algorithms

Number of TCP Connections	SS-PID	PID	PI
50	0.5	1.5	2
100	0.762834255	2.329006612	3.383951
150	1.156510035	3.466804857	4.451034
200	1.361514352	4.234558979	5.765755
250	2.621406441	4.753842241	6.475948

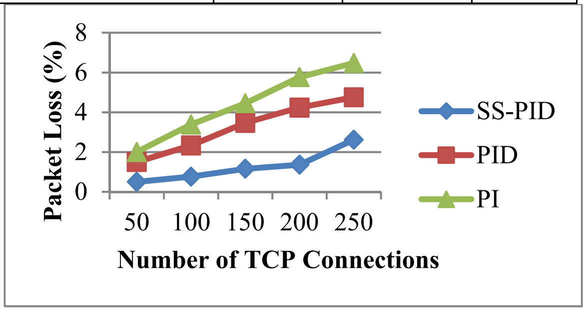

Fig. 6. Graphical representation of Number of TCP Connections Vs Packet Loss (%)

Afterwards, the Packet Loss is analyzed with respect to varying Round Trip propagation delay (ms). Here, the Packet Loss computed for the Round Trip propagation delay varied from 20, 40, 60, 80 and 100 is provided in table 4. Below table 4 clearly depicts that the packet loss is diminished when considering the Round Trip propagation delay also. Moreover, the minimal packet loss obtained for the proposed SS-PID shows its better performance efficiency than the conventional controller algorithms. Further, the graphical representation is also made in figure 7.

.

Table 4: Round Trip propagation delay (ms) Vs Packet Loss (%) for proposed and existing algorithms

Round Trip propagation delay (ms)	SS-PID	PID	PI
20	2	2.5	2.9
40	1.673900698	2.12511301	2.19648
60	1.220741456	1.744861887	1.881977
80	0.859646735	1.112096727	1.367379
100	0.147782436	0.382779234	1.02529

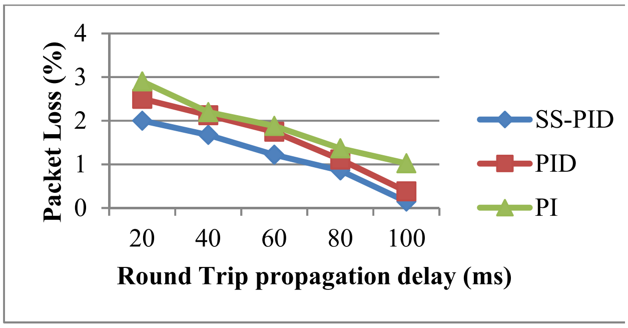

Fig. 7. Graphical representation of Round Trip propagation delay (ms) Vs Packet Loss (%)

4.1.4 Analysis made in terms of Link utilization (%) for varying Number of TCP Connections and Round Trip propagation delay (ms)

The performance metrics like, Link utilization is analyzed with varying Round Trip propagation delay (ms) and Number of TCP Connections in this section. The obtained values Link utilization (%) for varying Number of TCP Connections are tabulated in table 5 and graphically plotted in figure 8.

Table 5: Number of TCP Connections Vs Link utilization (%) for proposed and existing algorithms

Number of TCP Connections	SS-PID	PID	PI
50	0.85	0.87	0.9
100	0.851443137	0.874862	0.903987
150	0.855417137	0.882704	0.906561
200	0.857190622	0.884212	0.910916
250	0.860221607	0.888595	0.915445

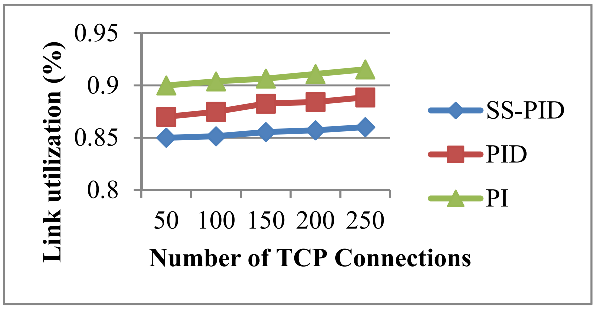

Fig. 8. Graphical representation of Number of TCP Connections Vs Link utilization (%)

Moreover, the values gotten for Link utilization measure when varying the Round Trip propagation delay (ms) values are given in table 6. Also, the graph is plotted for the gotten values in figure 9.

The values provided in both table 5 and 6 (i.e. on varying Number of TCP Connections as well as Round Trip propagation delay) illustrates that the link utilization is less for the proposed SS-PID when compared to the existing methods. But, the link utilization is more when number of TCP connections are increased and lowered when the Round Trip propagation delay is increased.

Table 6: Round Trip propagation delay (ms) Vs Link utilization (%) for proposed and existing algorithms

Round Trip propagation delay (ms)	SS-PID	PID	PI
20	0.82	0.91	0.98
40	0.807793156	0.902248	0.913212
60	0.716380421	0.803211	0.907875
80	0.645835142	0.741106	0.846361
100	0.630784212	0.647036	0.789753

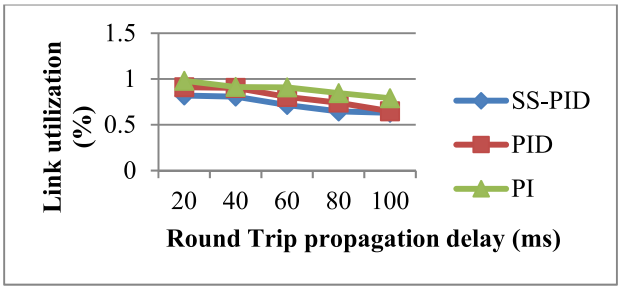

Fig. 9. Graphical representation of Round Trip propagation delay (ms) Vs Link utilization (%)

4.1.5 Analysis made in terms of Queue length (packet) for varying Time (ms)

In this section, the queue length values attained with respect to time is analyzed. The values gotten are provided in table 7, where the analysis is made on varying time metric from 0 to 9 ms. Moreover the graph is provided to show the accurate variations of the queue length values of the proposed SS-PID against the existing methods. Also, it is noted that the Queue length are lesser than the values gotten for conventional methods. Though, the values shows slight deviation with the target queue values, while the conventional methods results in higher deviation with the target queue length values in figure 10.

Table 7: Time (ms) Vs Queue length (packet) for proposed and existing algorithms

Time (ms)	SS-PID	PID	PI
0	2.76964	2.76964	2.76964
1	1314.64	2032.19	2471.2
2	2399.34	3397.46	3757.3
3	2403.37	3570.4	3292.91
4	2533.58	4064.32	4254.26
5	4034.86	4598.56	4681.66
6	3480.53	4664.55	4439.34
7	2987.26	4400.54	5037.71
8	3764.24	4592.65	4672.63
9	3468.36	4378.91	4811.86

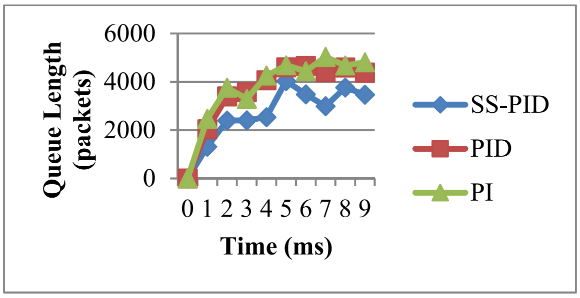

Fig. 10. Graphical representation of Time (ms) Vs Queue length (packet)

5. Conclusion

In this paper, we have introduced a Squirrel search based PID controller to dynamically find the controller gain parameters of PID controller based AQM in TCP networks. Here, the controller gain parameters are found by minimizing the integrated-absolute error (IAE) measure so that, the proposed controller design can guarantee less packet loss, stable queue length, and efficient link utilization for TCP networks. In order to show the efficiency of proposed SS-PID controller, we have analyzed using various performance metrics like delay, delivery ratio and queue length with respect to time; and then the packet loss and Link utilization by varying Number of TCP Connections and Round Trip propagation delay (ms) values. All the obtained values show better results of proposed SS-PID when compared to conventional PI and PID controllers. This defines the efficiency of proposed SS-PID controller design successfully.

References

1. Zhang, Wenli, and Yuanwei Jing. "Active Queue Management Algorithm Based on RBF Neural Network Controller." In 2020 Chinese Control And Decision Conference (CCDC), pp. 2289-2293. IEEE, 2020.
2. Zhang, Wenli, and Yuanwei Jing. "Active Queue Management Algorithm Based on RBF Neural Network Controller." In 2020 Chinese Control And Decision Conference (CCDC), pp. 2289-2293. IEEE, 2020.
3. Sharma, "A Survey on Active Queue Management Techniques", International Journal Of Engineering And Computer Science, 2016.
4. Awan, B. Ahmad and S. Ahmad, "Performance analysis of networks of queues under active queue management scheme", Simulation Modelling Practice and Theory, vol. 15, no. 4, pp. 416-425, 2007.
5. Hotchi, Ryosuke, Hosho Chibana, Takanori Iwai, and Ryogo Kubo. "Active Queue Management Supporting TCP Flows Using Disturbance Observer and Smith Predictor." IEEE Access 8 (2020): 173401-173413.
6. C. Chen, H. Kuo, J. Yan and T. Liao, "GA-based PID active queue management control design for a class of TCP communication networks", Expert Systems with Applications, vol. 36, no. 2, pp. 1903-1913, 2009.
7. Huang, Hongcheng, Gaofei Xue, Yuanming Wang, and Hongmei Zhang. "An adaptive active queue management algorithm." In 2013 3rd International Conference on Consumer Electronics, Communications and Networks, pp. 72-75. IEEE, 2013.
8. Wei, Tao, and Shun-yi Zhang. "Active queue management based on particle swarm optimization PID algorithm." In 2008 11th IEEE Singapore International Conference on Communication Systems, pp. 523-526. IEEE, 2008.
9. Rezaee and F. Pasandideh, "A Fuzzy Congestion Control Protocol Based on Active Queue Management in Wireless Sensor Networks with Medical Applications", Wireless Personal Communications, vol. 98, no. 1, pp. 815-842, 2017.
10. Sadek, Belamfedel Alaoui, Tissir El Houssaine, and Chaibi Noreddine. "A robust PID controller for active queue management framework in congested routers." In 2017 Intelligent Systems and Computer Vision (ISCV), pp. 1-6. IEEE, 2017.
11. S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance", IEEE/ACM Transactions on Networking, vol. 1, no. 4, pp. 397-413, 1993.
12. Xu, Qin, Fan Li, Jinsheng Sun, and Moshe Zukerman. "A new TCP/AQM system analysis." Journal of Network and Computer Applications 57 (2015): 43-60.
13. Wencong, Xu, Zhu Zijie, and Gao Xiang. "A Robust PID Non-fragile control theoretic analysis to TCP/AQM Network." In The

27th Chinese Control and Decision Conference (2015 CCDC), pp. 2188-2192. IEEE, 2015.

14. Hollot, V. Misra, D. Towsley and Wei-Bo Gong, "On designing improved controllers for AQM routers supporting TCP flows", Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213).
15. Hotchi, Ryosuke, and Ryogo Kubo. "Active queue management supporting TCP flows using dynamically controlled target queue length." In 2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), pp. 1-2. IEEE, 2018.
16. Chebli, Samira, Ahmed Elakkary, and Nacer Sefiani. "PID controller tuning using multi-objective ant colony optimization applied to TCP/AQM networks." In 2017 23rd International Conference on Automation and Computing (ICAC), pp. 1-6. IEEE, 2017.
17. Casoni, Maurizio, Carlo Augusto Grazia, Martin Klapez, and Natale Patriciello. "How to avoid TCP congestion without dropping packets: an effective AQM called PINK." Computer Communications 103 (2017): 49-60.
18. Liu, Yang, Xiaoping Liu, Yuanwei Jing, Ziye Zhang, and Xiangyong Chen. "Congestion tracking control for uncertain TCP/AQM network based on integral backstepping." ISA transactions 89 (2019): 131-138.
19. Sabry, Sana Sabah, Hadeel S. Obaid, and Atheer Marouf Al-Chalabi. "Grey Wolf Algorithm Based Pid Controller for Congestion Avoidance in Tcp Network." In 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), vol. 1, pp. 299-304. IEEE, 2019.
20. Bisoy, Sukant Kishoro, and Prasant Kumar Pattnaik. "Design of feedback controller for TCP/AQM networks." Engineering science and technology, an international journal 20, no. 1 (2017): 116-132.
21. Ma, Lujuan, Xiaoping Liu, Huanqing Wang, and Xiaoping Deng. "Congestion tracking control for multi-router TCP/AQM network based on integral backstepping." Computer Networks 175 (2020): 107278.
22. Wang, Kun, Yang Liu, Xiaoping Liu, Yuanwei Jing, and Georgi M. Dimirovski. "Study on TCP/AQM network congestion with adaptive neural network and barrier Lyapunov function." Neurocomputing 363 (2019): 27-34.
23. Wang, Kun, Yang Liu, Xiaoping Liu, Yuanwei Jing, and Siying Zhang. "Adaptive fuzzy funnel congestion control for TCP/AQM network." ISA transactions 95 (2019): 11-17.

Keerthipati Kumar is currently doing his research in the Department of Computer Science and Engineering, Vels Institute of Science, Technology and Advanced Studies, Chennai, India. He obtained his Master degree from Jawaharlal Nehru Technological University- Anantapur (2014). He has around 8 years of teaching experience and his areas of interest include networking, IOT, cloud computing**.**

Dr. R.A Karthika is currently working as Associate Professor, in the Department of Computer Science and Engineering, Vels Institute of Science, Technology and Advanced Studies, Chennai, India. She obtained her PhD degree from Noorul Islam University(2015). She has around 16 years of teaching experience and her areas of interest include networking, IOT, cloud computing, Big data. She has published around 45 papers in many national and international journals and also she is a member in various international accreditation bodies like CSI..