

# Development of an Enhanced Congestion Control Protocol in Transmission Control Protocol

Olumide Sunday Adewale  
Computer Science Department,  
Federal University of Technology,  
Akure

Muti Ganiyu  
Computer Science Department,  
Federal Polytechnic,  
Ile-Oluji

Olugbemiga Solomon Popoola  
Computer Science Department,  
Osun State College of Education,  
Ila-Orangun

Ibraheem Temitope Jimoh  
Computer Science Department,  
Federal University of Technology,  
Akure

## ABSTRACT

The common problems that motivate most of the researchers in the field of congestion control revolves around how to have a congestion free network while maintaining efficient links utilization, fairness, elimination of packet loss/drop as well as achieving throughput in an heterogeneous network environments. This work adopted internet prey-predator model protocol and enhanced it. The network topology consist of a router, an access point (AP), four wired and wireless nodes were used as test bed. Network Simulator 3 (ns-3) and C++ was used to simulate the work. The work examined the basis of congestion variant and develop a new enhanced congestion control protocol. The numerical results obtained under different simulation scenarios show that the new protocol manifest high bandwidth utilization, reduced loss rate, fairness among the flows and improved throughput. The new protocol is recommended to be deployed on the public transmission control protocol stack, this will not only improve the network performance but also better subscribers experience and give them value for their money. It will also afford the Internet Service Providers the opportunity to provide uninterrupted internet access. Future work can focus on improving the performance of TCP with chaotic control theory using machine learning to better explore the model.

## General Terms

Computer Network, Congestion Control, Predator-Prey Model, Lotka-Volterra, Algorithms.

## Keywords

Congestion Control, Transmission Control, Protocol, Enhanced, Network, Packet, Links, Fairness, Throughput.

## 1. INTRODUCTION

Congestion occurs when the rate at which packets are transmitted through a network begins to close to the packet handling capacity of the network. In queuing theory, traffic congestion is said to occur when the arrival rate exceeds the service rate of the system at a period of time. Congestion do occur at very high network traffic, performance may collapse completely, and all packets may not deliver. Bursty nature of traffic is the root cause of Congestion. Other factors, such as lack of bandwidth, ill-configuration and slow routers can also lead to congestion. Congestion control is a network layer issue, and is thus concerned with action taken when there are more data in the network that can be sent with reasonable packet delays at no packet loss.

Traditional Transmission Control Protocol (TCP) employs a network congestion-avoidance algorithm that includes various aspects of an additive increase/multiplicative decrease (AIMD) schemes. Scalable TCP (STCP) is a simple sender-side modification to the TCP congestion window update algorithm. It offers a robust mechanism to improve performance in high-speed wide area networks using traditional TCP receivers. STCP is designed to be incrementally deployable and behaves identically to traditional TCP stacks when small windows are sufficient, it uses MIMD algorithm. STCP alter the congestion window adjustment algorithm of the Standard TCP, the promptness with large windows can be dramatically improved. This work was carried-out using an enhanced nature inspired Internet Prey-Predator model (EIPPM) which adapt the work of Lotka-Volterra model published in 1885. The model is a non-linear differential equations that solve the problem of predator and prey relationship in life sciences [7]. EIPPM is also a chaotic algorithm which makes it suitable for resolving network congestion issues.

## 2. RESEARCH MOTIVATION

The common challenging problem that motivates most of the researchers in the field of congestion control revolves around how to have a congestion free network while efficient links utilization, fairness, elimination of packet loss/drop as well as maximum throughput is achievable in a heterogeneous network.

The limitation of [33, 12, 17, 35, 13] are the key motivation for this research work. These include:

- i. failure to consider loss rate using early congestion control (ECC) [33],
- ii. failure to considered related parameter such as loss rate, bit error rate (BER) and RTT [12],
- iii. failure to consider round trip time and heterogeneous network [17],
- iv. failure to consider wireless network environment [35],
- v. the need for cross layer bandwidth detection and dynamic control of *cwnd* to increase the accuracy of Bandwidth and handover estimations [13].

### 3. RESEARCH OBJECTIVES

The specific objectives of the research are to:

- i. design an enhanced Internet Predator-Prey Model (EIPPM) for scalable TCP that dynamically change congestion control protocols to better exploit the network capacities.
- ii. simulate the enhanced model.
- iii. evaluate the performance of proposed protocol

### 4. METHODOLOGY

In the food chain of ecosystem, it is assumed that  $x$  and  $y$  represent prey and predator respectively,  $N_x$  and  $N_y$  denote the number of each species.

The interactive relationships between  $x$  and  $y$  is explained in equations (1) and (2).

$$\frac{dN_x}{dt} = a_x \left(1 - \frac{N_x}{C_{N_x}}\right) - \alpha_y N_x N_y \quad 1$$

$$\frac{dN_y}{dt} = a_y N_x N_y - \alpha_{y_1} N_y \quad 2$$

Where  $\frac{dN_x}{dt}$  and  $\frac{dN_y}{dt}$  is the instantaneous rates of the prey and predators respectively,  $a_x$  is the natural growth rate of  $x$  and  $\alpha_y$  is the death rate of  $x$  when preyed upon by  $y$ .  $a_y$  represents the increasing rate of  $y$  after preying on  $x$ ,  $\alpha_{y_1}$  is the natural death rate of  $y$ .  $C_{N_x}$  refers to the carrying capacity of  $x$  in the absence of  $y$ , which means the maximum value of  $N_x$ . This is a two level predator-prey model and this research will extends this to three level model as shown in (3 – 5).

$$\frac{dN_x}{dt} = \left(a_x \left(1 - \frac{N_x}{C_{N_x}}\right)\right) - \alpha_{xy} N_x N_y - \alpha_{xz} N_x N_z - \alpha_x N_x^2 \quad 3$$

$$\frac{dN_y}{dt} = a_y N_y + \alpha_{yx} N_x N_y - \alpha_{y_1 z} N_y N_z - \alpha_{y_1} N_y^2 \quad 4$$

$$\frac{dN_z}{dt} = a_z N_z + \alpha_{zx} N_x N_z + \alpha_{zy} N_y N_z - \alpha_z N_z^2 \quad 5$$

$\frac{dN_x}{dt}$ ,  $\frac{dN_y}{dt}$  and  $\frac{dN_z}{dt}$  denotes the instantaneous rates of the bandwidth, packets and device update respectively.  $N_x$ ,  $N_y$ ,  $N_z$ , represents amount of bandwidth, packets and device updates respectively as function of time  $t$ ,  $a_i$  represents natural growth rates of three parameters;  $i = x, y, z$ .  $\alpha_x$  represents natural death rate of the bandwidth,  $\alpha_{xy}$  represents the rate of the bandwidth due to inhibition by the packet,  $\alpha_{xz}$  represents the rate of decrease of the bandwidth due to inhibition by the device updates.

$C_{N_x}$  represents bandwidth carrying capacity of the network link. In the three level predator-prey model, when the number of packets reduce sharply and device updates is not required, bandwidth will experience a logistics growth, so packets has an important role to reduce the growth of bandwidth. When the number of bandwidth reduces rapidly and substantially, packets transmission will likely decline and queue will grow exponentially on the network link and the model will sense the bustiness based on the queue size, it will dynamically reduce it by limiting the amount of incoming packets which either lead to packet drop or loss and this will be determined by the model based on the type of acknowledgement received by the sender, the model will also determine factors responsible for packet loss on the network.

The heterogeneous network is considered as an ecological system where lives species like *cwnd*, *queue length*, and *link capacity* [6, 7, 12]. The values of parameters represent the

number of each species which connected and influenced with each other, and the relationships between *cwnd*, available link capacity and queue length can be described as follows:

1. First, for available link capacity ( $N_x$ ) and *cwnd* ( $N_y$ ), when packet sending rate declines, available link capacity will grow and vice versa. On the other hand, if there is any growth of available link capacity, the sending rate will rise and vice versa. Moreover, the bandwidth in the bottleneck link also limits the maximum value of the available link capacity.
2. Second, for queue length ( $N_z$ ) and  $N_y$ , when lacking of waiting packet, *cwnd* will grow and vice versa. When packet sending rate declines, the queue length will decline and vice versa. The capacity of link limits the maximum value of packet sending rate.
3. It is obviously that the mutual restriction relations between  $N_x$ ,  $N_y$  and  $N_z$  conform to the logic of prey-predator model.

The heterogeneous network is considered as an ecological system where lives species like *cwnd*, queue length, and link capacity [6, 7, 12]. In other word, this work is using prey predator model to locate congestion problem in heterogeneous network, and the internet ecosystem is described in Figure 1.

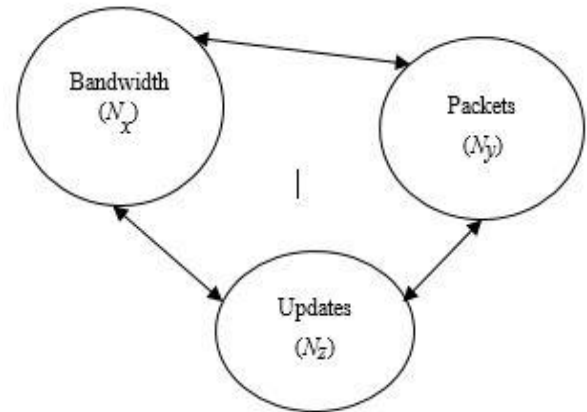


Fig. 1: Internet Ecosystem

Queue size is calculated using the equation (6), where the Packet size is 1040Byte

$$Queue\ Size = \frac{BB * RTT * 1000}{Packetsize * 8} \quad 6$$

#### 4.1 Congestion Control Algorithm

IPPM divide TCP congestion control into two phases: before packet loss and after packet loss. Preventing network congestion is the main task of IPPM before packet loss which was illustrate in [12].

After the loss one can distinguish the reason for packet loss and take the corresponding measures to deal with congestion loss and wireless loss, which is the part of contribution to knowledge of this paper.

As mentioned already in the previous paragraph, this work is designed as congestion control scheme that is not only reacting to loss but take necessary measures to prevent it, as this is the premier congestion feedback signal in the current Internet. Therefore, to be able to co-exist with existing Internet traffic, there is need to react to the same feedback signal, delay-based approaches usually react to an earlier congestion signal, namely increasing delay when the queue builds up, and as such are not able to allocate capacity when competing with loss-based traffic. By using delay

measurement to adapt the dynamic decrease/increase factor, Enhanced prey-predator protocol uses delay as a secondary congestion signal. Therefore, this protocol is a hybrid scheme.

Further, to frequently probe for newly available capacity and at the same time being able to quickly yield to link capacity, EIPPM is designed based on the TCP SIAD which is a variant of AIMD scheme. In corollary to this, EIPPM does not converge to a single steady state but operates in an equilibrium state between a minimum and maximum congestion window value as long as the network conditions are stable.

## 4.2 Standardized Metrics

The following are the standard metrics used in the research work:

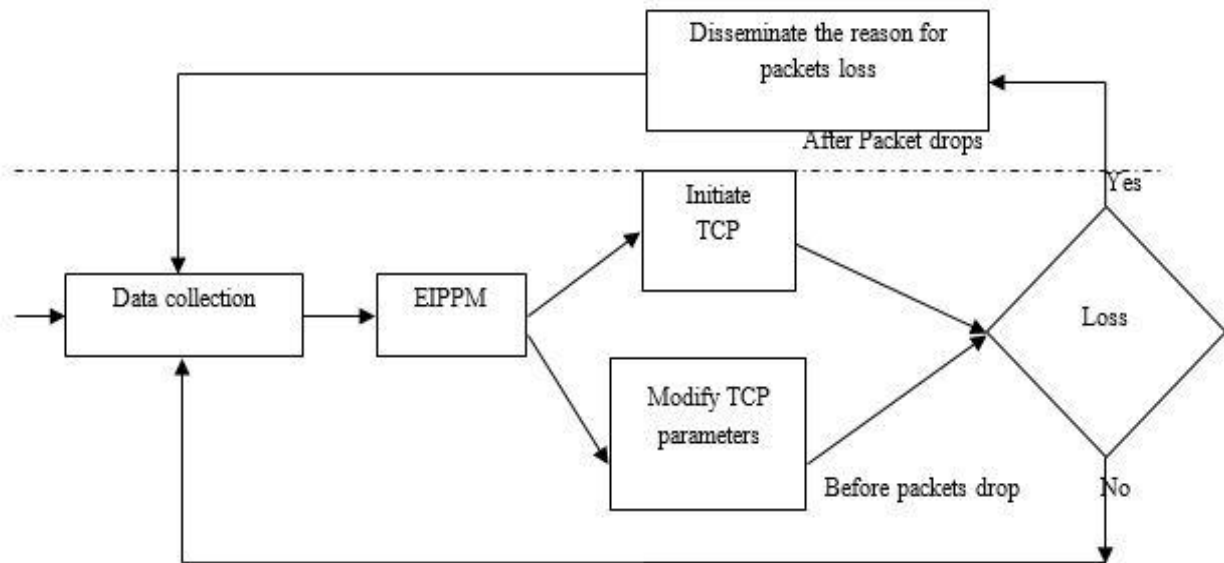


Fig. 2: The structure of EIPPM congestion control mechanism.

second or bytes per second. The number of jobs processed or successfully transmitted through/by the system per unit time.

$$\text{Throughput} = \text{Transfersize/Transfertime} = w * \text{MSS/RTT} \quad 7$$

### (b) Response time

Response time is the time required to receive a response to a request (round-trip time), the time taken for a very small packet to travel across the network and return. Delay is the time to traverse from one end to the other in a system. Delay (latency): In telecommunications there are several types of delay such as processing delay, propagation delay, queuing delay and transmission delay. In this thesis the notion of delay includes all the mentioned delay types and can be thus called end-to-end delay.

### (c) Goodput

Goodput is the rate in bits per second of useful traffic received. Goodput excludes duplicate packets and packets dropped along the path.

### (d) Jitter

Jitter is measures of the variability of the network delay time. This is expressed as a multiple of one standard deviation. The variation of packets' one-way delays is called variation (or jitter). The use of the term jitter is nowadays criticized as it has been used in different meanings by different groups. Swift packet delay variation (PDV) can be calculated from two

### (a) Throughput

Throughput is a measure of how much packets transferred across a link or a network in a certain time, it is usually measured in bits per

When TCP flow starts, the sender collect parameters from the network and compute a proper value of cwnd using Enhanced internet prey-predator model to initiate parameter dynamically. In this way, TCP flow can run with an optimum state and enhance the adaptability encountering different environments and transmission state, so as to avoid congestion occurrence as well as dropping packets. The processes of collecting and adjusting are conducted periodically, which improve the adaptability of TCP and take corresponding adjustment according to the dynamic change of network. The structure of proposed TCP congestion control algorithm is depicted as Figure 2.

successive packets' one-way delays. Delay variation can be caused by congestion in network, routing changes or timing drift.

### (e) Packet Loss/Dropped

Packet Loss/Dropped measures when one or more packets within a transmission are successfully sent, but fail to arrive at the destination.

## 5. RESULT AND DISCUSSION

This section presents the analysis of the existing system together with a detailed report on the design of the developed protocol. Every system that shows the characteristics of periodicity, strange attractor and initial value sensitivity, has been confirm to be a chaotic system. Interestingly, it has been established in [12] that TCP congestion control in heterogeneous network hold these characteristics using NS2. The network environment is shown in Figure 3, where, S denotes sender, AP (Access Point) represents the last jump of the wireless access point, D indicates the wireless receiver.

### 5.1 Strange Attractor of Chaotic Nature

Strange attractor is a production in chaotic system which reflects the motional characteristic of disorderly steady state, and it's also a kind of motion morphology of chaotic system. The chaotic nature of the system has been established through the analysis of the attractor of one dimension [6].

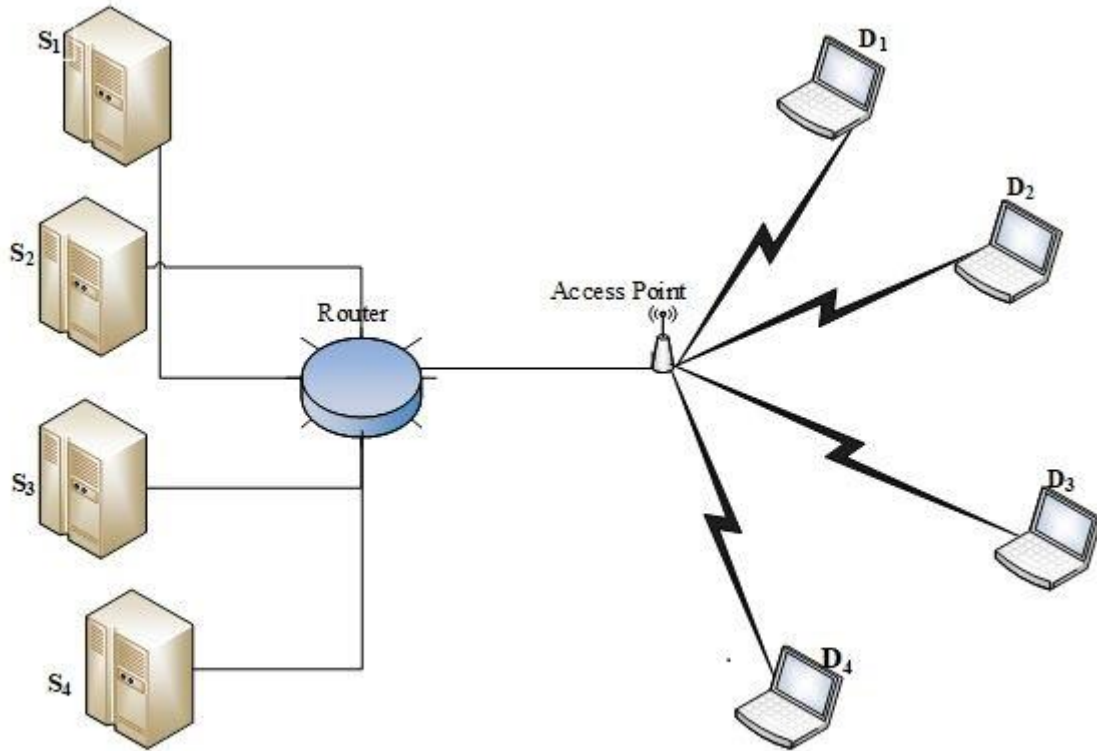


Fig. 3: The topological graph of Heterogeneous Network.

Because  $cwnd$  is just one of the numerous parameters in the whole system, and not continuous, which means that  $cwnd$  cannot describe the properties of a complex system effectively, especially the hidden nature of the system. It is particularly important to express the current operating condition of the system more accurately. A time shift algorithm proposed [28] to reconstruct the phase space of the parameter, so as to show some hidden properties in complex system effectively. The algorithm is shown as follows:

$$x[i] = \frac{1}{n} \sum_{j=1}^n cwnd_x[i - j] \quad 8$$

$$y[i] = \frac{1}{n} \sum_{j=1}^n cwnd_y[i - j] \quad 9$$

where  $x$  and  $y$  represent two TCP flows,  $n$  denotes the range of control statistics, a bigger value of  $n$  indicates a better reconstruction.

This method holds two advantages:

1) Let  $w$  represent the number of  $cwnd$ , the possible points of  $cwnd$  phase space will increase from  $w^2$  to  $(nw-n)^2$  using this method.

2) The distance between  $x[i]$  and  $x[i + 1]$  becomes smaller using this method, which should exactly not more than  $(2 * w - 2) / n$ . So this time shift algorithm can express the property of system more accurately and continuously. The initial conditions of systems are set as follow: the bandwidth of bottleneck link (BB) = 0.5Mbps, the link delay (LD) = 10ms, and queue length (QL) = 20packets, range of control statistics ( $n$ ) = 100. Data sends from  $S_1$  and  $S_2$  through wired link, and then they will be send from AP to  $D_1$  and  $D_2$  respectively. Two TCP flows are expressed as TCP0 and TCP1, the size of sending window is respectively labelled as  $cwnd(0)$  and  $cwnd(1)$ . After using the above algorithm to reconstruct the phases pace of  $cwnd$ , one can get Figure 4, where  $x[i]$  and  $y[i]$

represent a pair of values produced by time shift algorithm at a certain moment.

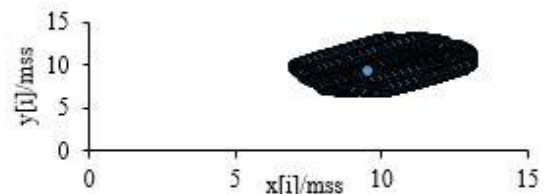


Fig. 4: The attractor in  $cwnd$  phase space.

Figure 4 indicates that the attractor appears after a sufficient running time, and it performs steady in some plausible disordered status. The common method to confirm an attractor as a strange attractor is to find out whether the attractor has fractal structure [38]. The fractal dimension of the attractor can be calculated by the method of box dimension. Assuming that  $S$  represents the square length of  $\Delta$ ,  $F$  is a non-empty and bounded set, and  $N(F)$  denotes the number of square where  $F$  intersects with  $S$ . Then the box dimension of above attractor  $D(F)$  can be expressed as follows:

$$D(F) = \lim_{\Delta \rightarrow \infty} \frac{\lg N(F)}{\lg(\frac{1}{\Delta})} \quad 10$$

$$y = kx + b \quad 11$$

In Eqs (10) and (11), the linear relation between  $y$  and  $x$  are  $y = \lg N(F)$  and  $x = \lg(1/\Delta)$  is considered.  $k$  is the slope of  $\lg N(F) \sim \lg(1/\Delta)$  and  $b$  is intercept. One can get the box dimension of this attractor  $D(F) = k \approx 4.44444$  after fitted by the least-square method. This box dimension is non-integer dimension, indicating that the attractor has a fractal structure which is termed strange attractor. The strange attractor is one

of the main characteristics of chaos, so it proves the essence of using the desired model.

## 5.2 Sensitivity to Initial Value of Chaotic Nature

The initial value sensitivity refers to a minor change at the beginning which results in a tremendous difference in the end. It is one of the important properties of chaotic system, called, butterfly effect. Next, let us analyze the initial value sensitivity of TCP congestion control in heterogeneous, considering the bandwidth of the bottleneck link (BB) = 2Mbps and the other conditions remain unchanged. Simulation results are compared among four groups of experiments. In each group, TCP0 begins to send data from 0.1s, while TCP1 starts from 0.1s, 0.5s, 1s and 5s respectively, both of them records the value of *cwnd* per 10ms. Experimental results are shown are graphically demonstrated in figure (5 – 8).

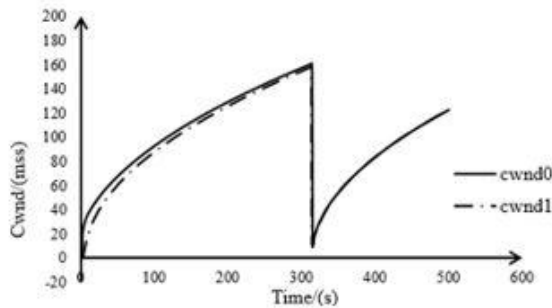


Fig. 5: Both TCP0 and TCP1 begins at 0.1s

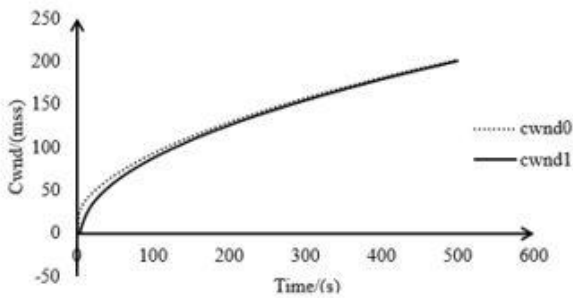


Fig. 6: TCP0 and TCP1 begins at 0.1s and 0.5s respectively

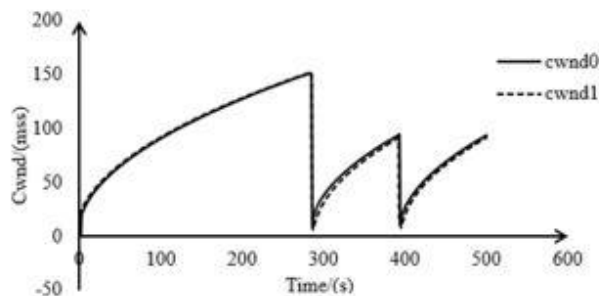


Fig. 7: TCP0 and TCP1 begins at 0.1s and 1s respectively

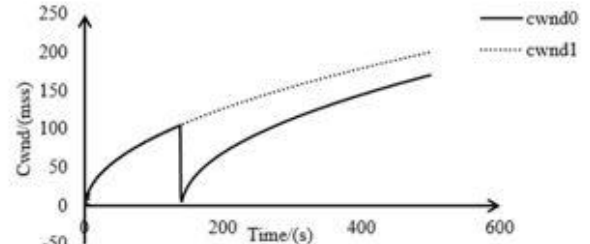


Fig. 8: TCP0 and TCP1 begins at 0.1 and 5s respectively

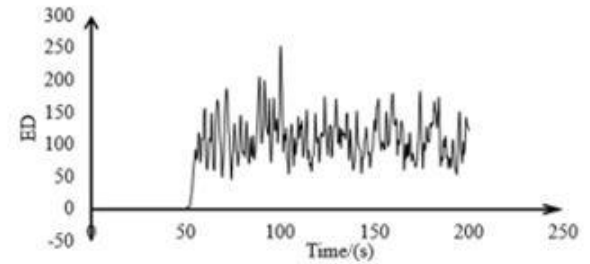


Fig. 9: The difference between the original and modify system

Figure 5 shows the changes of *cwnd* in the scenarios of two flows which starts at same period of time, the *cwnd1* stops at 300s and *cwnd0* experience tremendous packets loss even when there is no any changes to initial conditions. In a nutshell, the *cwnds* of two flows present irregularity and initial value sensitivity which meet the property of chaos system.

In the figure 6, it is evident that the system works well with no packet loss, and the two flows keep a high sending rate synchronously.

From figure 7, one can deduce that the two flows of the system experience packets loss almost the same and the two flows also converge at the same time.

Figure 8, one can deduce that one out of the two flows of the system experience packets loss almost the same and the two flows also converge at the same time. While in other cases, the system has a different degree of congestion or loss. Intuitively, the system in Figure 6 becomes simpler and efficient which can prove that a carefully chosen initial value can make a complex system change to a relatively simple and efficient system.

To further prove the initial value sensitivity of TCP in heterogeneous network, the number of TCP flows is increase to 30, specifically, BB = 1Mbps, LD = 15ms, QL = 60packets. When the running time arrived at 50s, one of the flows is choose randomly and its *cwnd* is increased by 1, then, compared the system with unchanged one. To show the differences between these two systems intuitively, this work used the Euclidean distance of *cwnd* in phases pace which states as follow:

$$Ed(t) = \sqrt{\sum_{i=1}^N (w^{orig}(i, t) - w^{pert}(i, t))^2} \quad 12$$

In equation (12),  $w^{orig}(i, t)$  represents the *cwnd* of the flow (*i*) at time *t* in the original system,  $w^{pert}(i, t)$  represents the value of *cwnd* of the flow *i* at time *t* in the changed system. *N* denotes the number of TCP flows and set to 30. Figure 9 shows the difference between two systems in phases pace of

*cwnd* over time. The Euclidean distance of two systems in the *cwnd*'s phase space is 0 before changed (before 50s), indicating that the two system is exactly the same. Due to the slightly change made at the 50s (increasing the *cwnd* of one flow), the subsequent evolution of the system is completely different compared with old value. It suggests that the system has the butterfly effect, and the butterfly effect is one of important index of the chaotic system. Given these features, it should come as no doubt that TCP congestion control has chaotic nature in heterogeneous network.

### 5.3 Analysis of Result

This section, analyzes and verify the changing trends of  $N_x$ ,  $N_y$  and  $N_z$ , then, evaluate the performance of EIPPM. Consider scenarios with four flows send from  $S_1 \sim S_4$  respectively in figure 3, and the initial values of *cwnd*s are set as:  $N_{y1} = 1$ ,  $N_{y2} = 6$ ,  $N_{y3} = 8$ ,  $N_{y4} = 13$ , specifically, other parameters are set as:  $\alpha = c_{N_x} = c_{N_y} = BB$ ,  $\alpha_x = 1$ ,  $\varepsilon = \alpha_y = \sigma = 0.1$ ,  $a_y = 0.5$ ,  $BB = 20\text{Mbps}$ ,  $LD = 30\text{ms}$ ,  $packet\ size = 1040\text{B}$ . The evaluation of the performance of throughput, fairness and packet loss rate of EIPPM is done in this section.

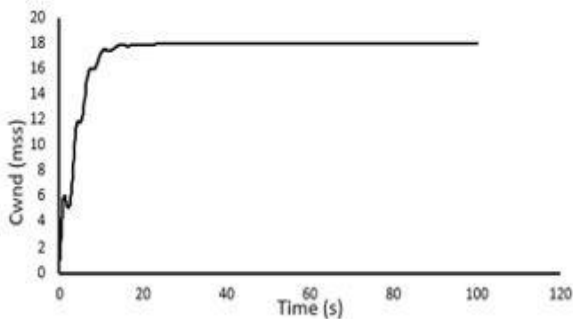


Fig. 10: The change of *cwnd* for  $N_{y1}$

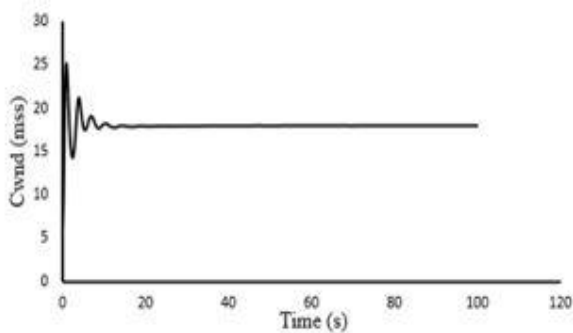


Fig. 11: The change of *cwnd* for  $N_{y1}$

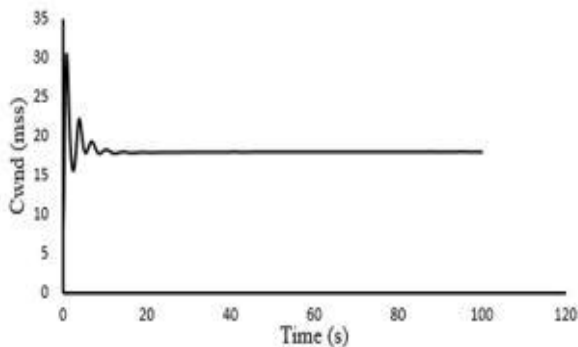


Fig. 12: The change of *cwnd* for  $N_{y3}$

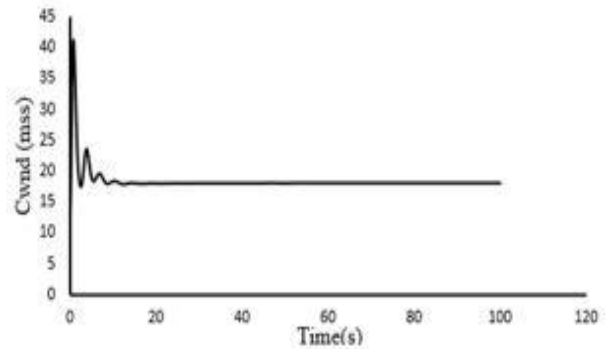


Fig. 13: The change of *cwnd* for  $N_{y4}$

From Figure 10 to Figure 13, one can concludes that even though  $N_{y1} \sim N_{y4}$  are set with different initial value, they can finally converge to a balanced state after running for a period of time, moreover, the sending rate of all flows are almost the same. So EIPPM is a cure for the traditional TCP with defects of unfairness.

To further establish the performance improvement of EIPPM protocol, its throughput is compared with traditional TCP under heterogeneous network and high bandwidth-delay product network, The work also evaluate the fairness and packet loss rate between proposed TCP and conventional TCP.

As it evident in Figure 14, the queue length of bottleneck link increases sharply at the first 5s then later achieves a steady-state that is tending to 0, which means there is no packets stacking in the bottleneck link and all packets are transmitted efficiently.

Figure 15 shows the change of available bandwidth of bottleneck link, which is almost near to 0 at about 30s, which means the utilization of bandwidth is almost up to 100%.

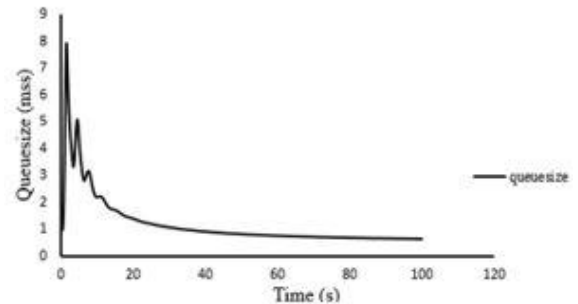


Fig. 14: Queue occurrence on the bottleneck link

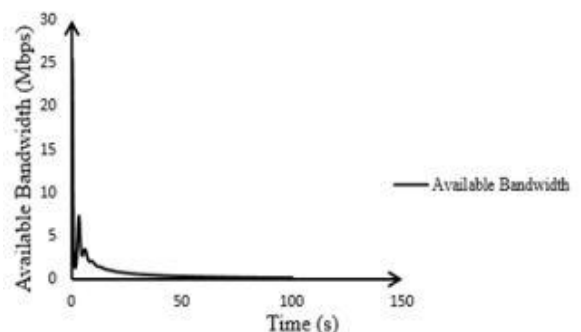


Fig. 15: Graph showing the available bandwidth

$$F(x) = \frac{\sum x_i^2}{m(\sum x_i^2)}$$

Where  $m$  denotes the number of users involved in computing the fairness index, and  $x_i$  represents the resource owned by user  $i$ . The value of fairness index is ranging from 0 to 1, the closer the index to 1, the better fairness of the performance. The fairness index of each flow is calculated using the above equation in the scenario of two flows. The fairness index of traditional TCP flow is 0.836. In contrast, the fairness index of new protocol flow is 0.997 which is almost close to 1. Obviously, the fairness of EIPPM protocol is much better than traditional TCP.

Furthermore, to evaluate the TCP fairness through different parameters, the work consider a 100M bottleneck link with round-trip delay (RTT) of 100ms. Simulation results are shown in Figure 18, it can be easily observed that the parameters converge to a steady state gradually under the constraint of EIPPM. More importantly,  $N_{y1}$  is overlap with  $N_{y2}$ , so there is no strong TCP or weak TCP, and two flows are allocated with fairly sending rate and bandwidth utilization.

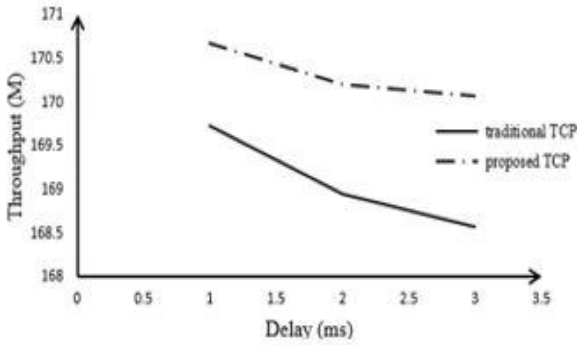


Fig. 16: Comparison of Throughput for 4 flows when bandwidth =20Mb

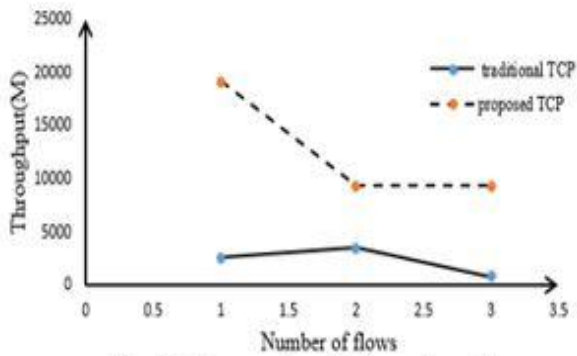


Fig. 17: Comparison of Throughput for high bandwidth-delay product network

1)Throughput under heterogeneous network: Figures 16 and 17 show the throughput comparison between new protocol and traditional TCP under scenario of 4 flows. As it is evident from these figures, throughput of new protocol is larger than the traditional TCP, and with the increment of bandwidth-delay product, the improvement of throughput is more outstanding.

2)Throughput under high bandwidth-delay product network: to eliminate the limitation of wireless and create a network with large bandwidth and long delay, the work choose the wired topology to simulate the experiment. The configurations simulated here are set as:  $BB = 100\text{Mbps}$ ,  $LD = 100\text{ms}$ . Figure 17 shows the throughput comparison between EIPPM and traditional TCP under scenarios of 2, 4 and 8 flows in large bandwidth-delay product environment. The throughput of EIPPM is 19156.68M, which is almost 6 times of the traditional TCP's 2582.36M.

Table 1: The comparison of throughput for 2 flows

Protocol Type	Throughput of Flow1 per Mb	Throughput of Flow2 per Mb
EIPPM protocol	9578.35	9578.33
Traditional TCP	999.90	1582.46

Table 1 shows that the throughput of flow 1 and flow 2 are almost the same in case of EIPPM protocol, which means these two flows share link resources equally while in traditional TCP, it is dramatically different.

3)Fairness performance: an equation named fairness index as shown in 10 is introduced.

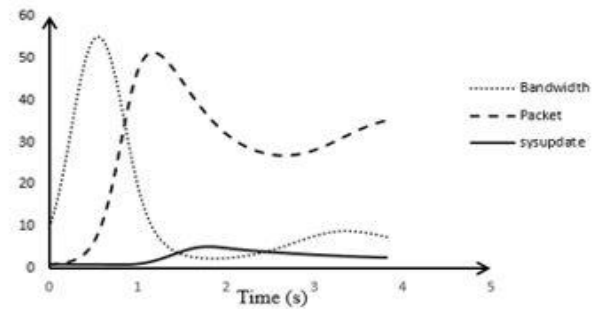


Fig.18: The fairness of the flows

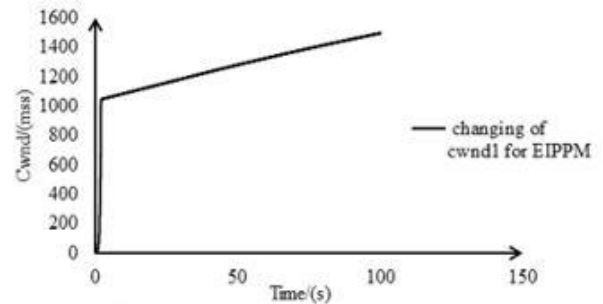


Fig. 19: The changing of cwnds for traditional TCP

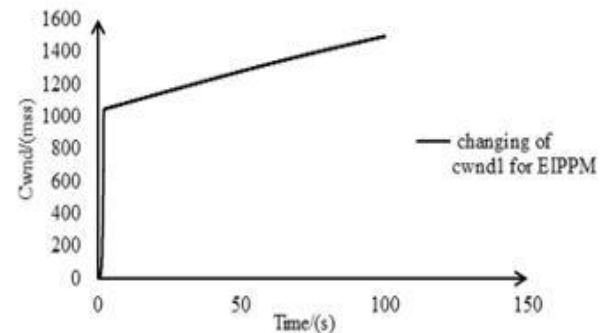


Fig. 20: The changing of cwnds for EIPPM

4) Packet loss performance: Figure 19 shows the changing of *cwnds* under the traditional TCP, one can find that  $N_{y1}$  encounters packet loss at 20.8s and  $N_{y2}$  loss packet at 23.2s. After packets loss, senders cut down their *cwnd* to 1, then stay in congestion avoidance phase until the end of simulation. Unfortunately, the sending rates of two flows maintain at a low level as well as the bandwidth utilization during the stage of congestion avoidance.

Considering Figure 20, the *cwnds* of these two flows increase to 1000 exponentially at first 4s and step into congestion avoidance phase until it finished with the sending of packets. There is no packet loss and the sending rate remains at a high level during simulation.

In simulations of traditional TCP, when the first packet loss happens, the high sending rate will lead to consecutive packets losses. As a result, senders will re-transmit the dropping packet and remain at a small sending rate in the congestion avoidance phase, and that's the reason why traditional TCP performance degraded significantly. Since EIPPM protocol could prevent the packet loss, it could improve the performance of conventional TCP. Given these features, there should be no doubt that a significant contribution to improve the performance of conventional TCP in reducing the packet loss has been made by EIPPM.

As stated earlier, *ssthresh* is the threshold of *cwnd* between slow start and congestion avoidance phase, where the *cwnd* will increase exponentially and later turn to logistic growth. Therefore, chosen an appropriate value for *ssthresh* is very important. This is the reason *ssthresh* is chosen as the parameter to be initialized so as to improve the performance of conventional TCP. Experiments were carried-out using topology shown in Figure 3, and the results are compared with the traditional TCP. The following scenarios are considered where *Bandwidth* = (10, 20Mbps), *RTT* = (10, 20, 30ms), and the number of flows = (2, 4, 8). The values of *ssthresh* are computed by EIPPM protocol in different conditions are shown in Table 2 and 3.

**Table 2: *ssthresh* value computed by EIPPM for 10Mb**

Delay (mb)	Flow Number	Ssthresh Value
10	2	6.00
	4	3.01
	8	1.50
20	2	11.5
	4	6.01
	8	3.00
30	2	16.38
	4	9.01
	8	4.50

**Table 3: *ssthresh* value computed by EIPPM for 20Mb**

Delay (mb)	Flow number	Ssthresh Value
10	2	12.00
	4	6.01

	8	3.00
20	2	23.00
	4	12.03
	8	6.00
30	2	34.32
	4	18.02
	8	9.00

From the table 2 and 3, it is evident from *ssthresh* values that congestion control mechanisms such as Traffic shaping, Admission control, Quality negotiation as well as node shedding are the key factor to consider if one truly want to have desirable congestion windows that will result in high bandwidth utilization and best throughput on the network.

#### 5.4 The Protocol as Network Backbone

The total packets to be transmitted were set to be = 7701M while other parameters remain unchanged and the following results are gotten:

- i. 7678M of packets were received while 23M of packets were loss due to internal queue and longtime delay when system updates is not required by the devices on the network.
- ii. 7656M of packets were received while 45M of packets were lost due to internal queue and longtime delay when system updates was running concurrently without denying the user the resources paid for.
- iii. Average Goodput is 8.83048Mbps while 9.9646Mbps accounted for throughput.
- iv. Mean Jitter value is 0.00119.

#### 6. CONCLUSION

This work examined the basis of congestion control in traditional and scalable transmission control protocol, different congestion control mechanism and algorithm of transmission control protocol variants, the existing literatures on the structure of traditional and scalable TCP congestion control protocol combined with the merit of nature-inspired algorithm, which can initiate the TCP flexibly using internet prey-predator model of food chain in biological science and makes the network converge to a stable state and achieve a balanced state with high bandwidth utilization within short period. Numerical results in different simulation shows that the new protocol manifests high bandwidth utilization, improved round throughput, reduced loss rate and fairness. This new protocol variant is recommended to be deployed on public transmission control protocol stack, it will not only improves the network performance but also gives end users or customers value for their money and provide uninterrupted network access to the public that devoid of existing network operational architecture that do forced the Internet Service Provider (ISP) to schedule time for network updates at the expense of denying the user the internet access during the period of the updates. An interesting direction for further work can focus on improving the performance of TCP with chaotic control theory using machine learning to better explore the model.



## 7. REFERENCES

- [1] Afanasyev, A., Tilley, N., Reiher, P., & Kleinrock, L. (2010). Host-to-host congestion control for TCP. *IEEE Communications surveys & tutorials*, 12(3), Pages 304-342.
- [2] Ahmad, S. S. (2016). Survey on Different Types of TCP Protocols for High Speed Networks. *International Journal of Advance Research in Computer Science and Management Studies*, Volume 4 Issue 6, pages 122-125.
- [3] Allman, M. (2003). TCP Congestion Control with Appropriate Byte Counting (ABC)", RFC 3465, retrieved from <https://tools.ietf.org/pdf/rfc3465.pdf>.
- [4] Allman, M., Paxson, V., & Stevens, W. (1999). TCP congestion control, retrieved from <https://www.ietf.org/rfc/rfc5681.txt>.
- [5] Alizadeh, M., Greenberg, A., Maltz, D. A., Padhye, J., Patel, P., Prabhakar, B. & Sridharan, M. (2011). Data center tcp (dctcp). *ACM SIGCOMM computer communication review*, 41(4), 63-74.
- [6] Awrejcewicz, J. (2014). *Chaos and Synchronization. In Ordinary Differential Equations and Mechanical Systems*, Springer, Cham, Page 527-604.
- [7] Das K., Reddy K. S., Srinivas M.N., Gazi N.H. (2014), Chaotic Dynamics of a Three Species Prey–Predator Competition Model with Noise in Ecology. *Journal of Applied Mathematics and Computation* 231 pages 117–133.
- [8] De Schepper, K., Bondarenko, O., Tsang, I. J., & Briscoe, B. (2016, December). PI 2: a linearized AQM for both classic and scalable TCP. In *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies* (pp. 105-119).
- [9] Floyd, S., Podolsky, M., Mahdavi, J., & Mathis, M. (2000). *An extension to the selective acknowledgement (SACK) option for TCP*, retrieved from <http://www.hjp.at/doc/rfc/rfc2883.html>.
- [10] Ha, S., Rhee, I., & Xu, L. (2008). CUBIC: a new TCP-friendly high-speed TCP variant. *ACM SIGOPS operating systems review*, 42(5), Pages 64-74.
- [11] Hayes, D., Tsang, J., Ros, D., Petlund, A., & Briscoe, B. Internet Latency: Causes, Solutions and Trade-offs. *2015 European Conference on Networks and Communications (EuCNC)*, pages 545-552.
- [12] Hui, W., Peiyu, L., Zhihui, F., Zheqing, L., & Xuhui, W. (2016), Enhancing the Flexibility of TCP in Heterogeneous Network. *PloS one*, 11(9), e0161249.
- [13] Kanagarathinam, M. R., Singh, S., Sandeep, I., Roy, A., & Saxena, N. (2018). D-TCP: Dynamic TCP congestion control algorithm for next generation mobile networks. In *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Pages 1-6.
- [14] King, R., Baraniuk, R., & Riedi, R. (2005, March). TCP-Africa: An adaptive and fair rapid increase rule for scalable TCP. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*. Vol. 3, Pages 1838-1848.
- [15] Konda, V., & Kaur, J. (2009, April). RAPID: Shrinking the congestion-control timescale. In *IEEE INFOCOM 2009*, Pages 1-9.
- [16] Kotian, P. P., Shetty, V. K., & Begum, S. (2017). Study on Different Mechanism for Congestion Control in Real Time Traffic for MANETS. *International Research Journal of Engineering and Technology (IRJET)*, Volume 04 Issue 11, Page 1627-1631.
- [17] Kushwaha, V. (2016). Interaction of High Speed TCPs with Recent AQMs through Experimental Evaluation. *International Journal of Computer Network & Information Security*, 8(9). pages 41-47.
- [18] Kühlewind, M. (2013). Adaptive and Scalable Congestion Control wanted!. In *ISOC Workshop on Reducing Internet Latency*. [https://www.internetsociety.org/wp-content/uploads/2013/09/21\\_bis\\_adaptive\\_cc\\_shorten.pdf](https://www.internetsociety.org/wp-content/uploads/2013/09/21_bis_adaptive_cc_shorten.pdf)
- [19] Kühlewind, M. (2016). *Scalable increase adaptive decrease: congestion control supporting low latency and high speed*. Universität Stuttgart, Institut für Kommunikationsnetze und Rechnersysteme.
- [20] Kühlewind, M., & Briscoe, B. (2010). Chirping for Congestion Control-Implementation Feasibility. *Proceedings of PFLDNeT'10*.
- [21] Leong, W. K., Wang, Z., & Leong, B. (2017, November). Tcp congestion control beyond bandwidth-delay product for mobile cellular networks. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*. Pages 167-179.
- [22] Liu, S., Vojnovic, M., & Gunawardena, D. (2007). Competitive and considerate congestion control for bulk data transfers. In *2007 Fifteenth IEEE International Workshop on Quality of Service*, pages 1-9.
- [23] Mangiante, S., Schapira, M., Navon, A., Silva, M. D., Godfrey, B., Wang, W., & Pechtalt, I. (2018, October). Congestion Control for Future Mobile Networks. In *Proceedings of the 13th Workshop on Challenged Networks* (pp. 55-61).
- [24] Mathis, M. (2009). Relentless congestion control. In *Proc. PFLDNeT*. <https://www.gdt.id.au/~gdt/presentations/2010-07-06-questnet-tcp/reference-materials/papers/mathis-relentless-congestion-control.pdf>
- [25] Mathis, M., Dukkupati, N., & Cheng, Y. (2013). Proportional rate reduction for TCP. Internet Engineering Task Force (IETF), RFC, 6937, Retrieve from <http://www.hjp.at/doc/rfc/rfc6937.html>.
- [26] Nguyen, T. A. N., Gangadhar, S., & Sterbenz, J. P. (2016, June). Performance evaluation of TCP congestion control algorithms in data center networks. In *Proceedings of the 11th International Conference on Future Internet Technologies*, pages 21-28.
- [27] Nithya, B., Mala, C., & Sivasankar, E. (2014). A novel cross layer approach to enhance QoS performance in multihop adhoc networks. In *2014 17th International Conference on Network-Based Information Systems* (pp. 229-236).

- [28] Packard, N. H., Crutchfield, J. P., Farmer, J. D., & Shaw, R. S. (1980). Geometry from a time series. *Physical review letters*, 45(9), Page 712.
- [29] Padhye, J., Firoiu, V., Towsley, D., & Kurose, J. (1998). Modeling TCP throughput: A simple model and its empirical validation. *ACM SIGCOMM Computer Communication Review*, 28(4), 303-314.
- [30] Padhye, J., Firoiu, V., Towsley, D. F., & Kurose, J. F. (2000). Modeling TCP Reno performance: a simple model and its empirical validation. *IEEE/ACM transactions on Networking*, 8(2), Pages 133-145.
- [31] Rao, V. P., Tahiliani, M. P., & Shenoy, U. K. K. (2014). Analysis of sfqCoDel for active queue management. *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)* pages 262-267.
- [32] Rhee I. and Xu L. (2005). CUBIC: A New TCP-Friendly High-Speed TCP Variant. [www4.ncsu.edu/rhee/export/bitcp/cubic-paper.pdf](http://www4.ncsu.edu/rhee/export/bitcp/cubic-paper.pdf).
- [33] Ribeiro, V. J., Riedi, R. H., Baraniuk, R. G., Navratil, J., & Cottrell, L. (2003,). pathchirp: Efficient available bandwidth estimation for network paths. In Passive and active measurement workshop, retrieved from <https://scholarship.rice.edu/bitstream/handle/1911/20252/Rib2003Apr5pathChirp.PDF?sequence=1&isAllowed=y>.
- [34] Sathiaseelan, A., & Radzik, T. (2005, April). Robust TCP (TCP-R) with explicit packet drop notification (EPDN) for satellite networks. In *International Conference on Networking*, pages 250-257.
- [35] Sánchez, J. F., & Cuellar, M. M. (2017), Proposal of a congestion control technique in LAN networks using an econometric model ARIMA. *Advances in Science, Technology and Engineering Systems Journal* Vol. 2, No. 1, pages 269-276.
- [36] Shalunov, S., Hazel, G., Iyengar, J., & Kuehlewind, M. (2012). Low extra delay background transport (LEDBAT). *draft-ietf-ledbat-congestion-04.txt*.
- [37] Shimonishi, H., Hama, T., & Murase, T. (2006, February). *TCP-adaptive reno for improving efficiency-friendliness tradeoffs of TCP congestion control algorithm*. In *Proc. PFLDnet*, pages 87-91.
- [38] Shivamoggi B. K. (2014), *Chaos in Dissipative Systems*. Springer Netherlands, v103, Pages 189–244.
- [39] Talau, M., Fonseca, M., Munaretto, A., & Wille, E. C. (2016, November). Early congestion control: A new approach to improve the performance of TCP in ad hoc networks. In *2016 7th International Conference on the Network of the Future (NOF)*, pages 1-6
- [40] Tan, K., Song, J., Zhang, Q., & Sridharan, M. (2006, April). A compound TCP approach for high-speed and long distance networks. In *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications* Pages 1-12.
- [41] VijiPriya J. and Suppiah S. (2016), *An Extended Study On Newton Raphson Congestion Control*. *International Journal of Advanced Research*, Volume 4, Issue 2, pages 574-58.
- [42] Yadav A. and Kumar S. (2018), Novel additive increase and multiplicative decrease algorithm for congestion control in TCP. *International Journal of Pure and Applied Mathematics*, Volume 118 (20), Pages 1059-1066.
- [43] Zarchy, D., Mittal, R., Schapira, M., & Shenker, S. (2017, November). An axiomatic approach to congestion control. In *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, Pages 115-12
- [44] Zhang, W., Lin, L., & Du, L. (2017). The study of secure congestion control for TCP in Ad hoc networks. *Journal of Information Security*, Volume 9 Issue 1, pages 25-32.