Kernel Network Parameters Optimization using Optimized Artificial Neural Network to Serve Quality of Service Parameters

Hiteshkumar M. Nimbark
JJT University, Jhunjhunu, Rajasthan, India

Dr. P P Kotak
Professor/Principal, Gujarat Technological University, Gujarat

Abstract- The High-performance computer networks (HPNs) is major important in the success of critical mission, computer network centric operations when data volumes and near real-time data processing requirements is very high. Now a days the data requirement is increase in 3 to 5 folds. Based on Lowest time latency, big enough bandwidth with high steadfastness, the High performance network is benefited a lot. The applications and process like replication and recovery of datacenter and distributed computing network require support of very big transfer and very small network latency time to perform the mission of critical requirement. Ordinally and Traditionally there are two main approaches deal with the same for justify the Quality of Service(QoS). The first is to create for each service request a reservation state and keep it for all the connection known as Integrated Service Architecture (IntServ). The another one is to carry out the categorization of the traffic at the ingoing of the network. The policy administration created then after is known as Differentiation Service (DiffServ). We here with proving the Artificial Neural Network based operating system Kernel network parameter optimization approach. With an accurate characterization in hand, we have then proposes kernel modification and overhauls to the TCP protocol to significantly improve network performance. We also have optimized neural network using Artificial Bee Colony Algorithm (ABC) to balance between exploration and exploitation characteristics of Neural Network. This is the novel approach in the field of High Performance Network and QoS. During the study of the computer network traffic the accurate of the classification and characterization is still been untouched due to diversify focused of researchers. There is need to more rigorous study of network traffic as we are entering head into the Next-Generation Internet (NGI).

Keywords - Artificial Bee Colony Algorithm, Nature-Inspired Meta-heuristics, Optimizations, TCP, Artificial Neural Network

I. INTRODUCTION

Since 1994, research has shown that aggregate network traffic can be characterized as busty, or more specifically, self-similar or fractal. However, there has been only limited work on understanding why the traffic behavior is self-similar. While heavy-tailed distributions in file size, packet inter-arrival, and transfer durations may contribute to the self-similarity, we have found that the primary source of self-similarity is from the protocol stack itself, namely TCP. More specifically, the self-similarity is due to a particular implementation of TCP - TCP Reno, a ubiquitously deployed transport protocol found in virtually all modern operating systems.

The Quality of Service (QoS) is that it means providing consistent, predictable data delivery service. It covers a diverse set of service. It is known as secutity, availability, reliability and performance. The need of national high performance network infrastructure is on high tone which include bandwidth and accessibility there are still small no of users actually take full advantages of the available infrstrature. Without expert involvement from network engineers, users are unlikely to get 10 Mbps solitary stream TCP transfers, though the fundamental network infrastructure is capable to support data rates of 1000Mbps or more. This problem categoriesed in two factors: host operating system or system software that is optimized for low bandwidth, and the lack of efficient tools to diagnose performance. The congestion control algorithms [1] and large window extensions in TCP permit a host running a single TCP stack to support concurrent connections across the entire range of bandwidth. Basically all application that use TCP should be able to share of available bandwidth. That is also on any path without concerning manual configuration. In this paper work we proposes a system for adaptive Neural Neetwork based instrumentation of kernel parameters sizes based upon network conditions and system memory availability. It is intended to operate transparently without modifying existing applications. Since it does not change TCP's basic characteristics, it does not change TCP's basic interactions with the Internet or other Internet applications. The kernel instrument set extended kernel of project RFC4898 will be utilized to address the problem with new proposed Neural Network

approach. Prototypes of these instruments will be implemented in Linux Kernel 3.11 This is working to provide an instrumented implementation of TCP as part of standard Linux. This implementation will expose TCP's hidden machinery and allow neural network based TCP support centre to optimized QoS parameters. In sub sequent topics we will cover the details of how the Artificial Neural Network can be utilized to optimized kernel parameters.

II. CO-ABC BASED ARTIFICIAL NEURAL NETWORK OPTIMIZATION AND RECOMENDATION

Artificial Bee Colony (ABC) algorithms was design to use unconstrained benchmark optimization[2]. Compare to other well-known meta-heuristic algorithms the extended Version of the ABC algorithm is introduced to handle many optimization problems. Furthermore ABC result re compared with other optimization algorithm from all survey we found ABC is best still there are some limitation that is known as exploration and exploitation balancing. ABC is good at exploration but poor at exploitation. For this limitation is removed by using modified version of ABC [3].

Artificial Neural Network is appearing to be a recent development. Currently, the neural network field emphasis a resurgence of interest and a corresponding increase in researchers. A trained neural network can be thought of as an professional experts in the category of knowledge it has been given to examin or study. This expert can then be used to provide project the available new situations. It is obviously self real time organization and adaptive learning. It can be use to Fault Tolerance Coding makes it to utilized in many latest real time queries to solve problems. We use designed neural network using (Chaos Opposition)CO-ABC algorithms to perform the kernel parameter optimization.

The ABC algorithm [3] has a strong ability to find good at exploration but poor at exploitation, but this problem can be removed by using proposed chaos-opposition based initialization instead of random initialization population. Main idea behind this approach is used at the beginning stage of searching for the optimum. The basic ABC we are not discussing over here becase of space related issues we will discuss directly the use of CO-ABC for ANN design and recommendation process. ANN will constructed through following phases:

Initializing Population: Population initialization is a crucial task in meta heuristics algorithms because it can affect the convergence speed and the quality of the final solution. CO-ABC initialization approach which employs opposition-based learning method and chaotic systems to generate initial population. Here, sinusoidal iterates is selected and its equation is defined as follows:

$$X_{best,j} = X_{min,j} + ch_{k,j}(x_{max,j} - x_{min,j}) \qquad ch_{k+1} = sin(\pi ch_k), ch_k \epsilon (0,1), k = 0,1,...,K \quad \ldots(1)$$

Where k is the iteration counter and K is the present maximum number of chaotic iterations. The variables in Eq.1 can distribution in search space with sinusoidal periodicity, randomness and irregularity. The best state solution in current state is very useful and directly affect the movement of current population. The solution search equation is devised as follows:

Looping with Fitness Function:

$$V_{ij} = X_{ij} + \Phi_{ij}(X_{ij} - X_{kj}) \quad \text{-------- (2)} \qquad ABC = best = 1 : V_{ij} = x_{best,j} + \phi_{i,j}(x_{r1,j} - x_{r2,j}) \quad \text{------ (3)}$$

Where the indices r1 and r2 are mutually exclusive integers randomly chosen from 1,2, ... ,SN (node) and different from the base index i; X_{best} is the best individual vector with the best fitness in the current population and J=1,2, . . . ,n is randomly chosen indexes; $Ø_{i,j}$ is a random number in the range [-1,1]. In Eq.2, original ABC initialization and looping process the coefficient $Ø_{i,j}$ is a uniform random number in [-1, 1] and $X_{k,j}$ is a random individual in the population. Therefore, the solution search dominated by Eq.2 is random enough for exploration. In other words, the solution search equation described by Eq.2 is good at exploration but poor at exploitation. However, according to proposed CO-ABC initialization and looping Eq.3, ABC/best/1 can drive the new candidate solution only around the best solution of the previous iteration. Therefore, the proposed solution search equation described by Eq. 3 can increase the exploitation of ABC. When the weight and bias of an individual is updated by means of the MSE function or know as fitness function[19]. For the case of the classification error function given in eq.5

$$F_1 = \frac{1}{N}\Sigma_{i=1}^{n}(T_i - O_i)^2 \quad \text{---- (4)} \qquad F_2 = 1 - \frac{(npwc)}{tnp} \quad \text{---- (5)}$$

Where, npwc is the number of patterns to be classified and tnp is the total number of patterns. Now, F1 and F2 function that not help to minimize the number of connections of the ANN, F3 used for minimize number of connections. Where, NC is the number of connections in the ANN designed by the Proposed CO-ABC methodology

and NMaxC is the maximum number of connections generated with neuron. Proposed two new functions are From the F1, F2 & F3

$$F_3 = \frac{NC}{NMaxC} \quad \text{---- (6)} \qquad \begin{aligned} FF_1 &= F_1 * F_3 \\ FF_2 &= F_2 * F_3 \end{aligned} \quad \text{---- (7)}$$

These functions are optimize using CO-ABC algorithm.

Transfer Function: Here, define one nonlinear activation function is radial basis function is

$$O = \exp(I)$$

$$Where, I = \frac{-\Sigma_{i=1}^{n}(W_i(t) - X_i(t))^2)}{2\sigma^2} \quad \text{.....(8)}$$

Recommendation: ANN use an algorithmic approach while normal computer follows a set of instructions to solve the specific problem. The conventional computer use known steps to solve the problem can restricts the problem solving capability of conventional computers. Computers can be so much useful if it could do things that we don't exactly know how to do. Neural networks process information in a similar way the human mind process. There are many recommendation systems use for predicting the possibilities like a Probabilistic Approach, The Naïve Bayes Classifier, Non-probabilistic approach Support Vector Machines, A Metric-based Classifier: K-Nearest Neighbors, A Non-Metric Classifier: Decision Trees, Multilayer Neural Networks. For our project the Neural Network is best suitable as we are optimizing the design of ANN for optimizing QoS parameters. We will use V_{ij} as ABC/BEST/1 matrix to recommend from the data set with best MSE or Fitness function values.

III. TCP WINDOWING, THROUGHPUT AND WEB10G

A TCP window of data is the amount of data transmitted on a connection during one round-trip time. The concepts regarding various TCP windows used here, is explained below. Receive window: is the buffer amount as receiving end to store data before process or send it upper layer of TCP stack. TCP implements flow control by announcing a receive window (rcWnd) in each segment header. This window announced by the receiver is an upper bound on the sender's window size. If the application reads slower than data arrives, the buffer will start to fill. As the buffer fills, the window size and therefore throughput will fall (eventually reaching zero when the buffer fills entirely). Throughput will quickly match the application's consumption rate. same way Retransmit queue affect the throughput. In many cases, the amount of traffic coming into a router is greater than its outgoing bandwidth. A finite quantity of packets may be queued in the router, but in a steady state a certain percentage must be dropped. In this case, the router is said to be a bottleneck and experiencing congestion. One of TCP's functions is to detect this congestion (usually by observing lost packets) and limit its transmission rate accordingly. A TCP sender maintains a congestion window, an upper bound on window size based on the observed properties of the network. Same way Sending window affect the throughput. we ideally have: window = cwnd and throughput = bandwidth (BW). Since throughput = window/Round Trip Time (RTT), we have bandwidth = cwnd/RTT. Therefore, we have:

$$cwnd = BW * RTT \quad \text{---------(9)}$$

This quantity is known as the bandwidth-delay product (BDP). Consequently, BDPs across the Internet are also increasing over time. Further, connection BDPs may vary by many orders of magnitude on a single host. For example, a host connected with 100 Mbps fast Ethernet may connect to a similar host at the same time as it connects to one with a 56 kbps modem. Mathis [4] described the relationship between the upper bound of TCP bandwidth, packet round trip time RTT , and packet loss p with the equation 10

$$BW \leqslant (MSS/RTT) * (C/\sqrt{p}) \quad \text{---- (10).}$$

We are considering c=0.7 as per guided by Matt Mathis [4]. To achieve substantial network performance over a wide area network that has a relatively large RTT, the required maximum packet loss rate p must be very low. The relationship derived by Mathis for the maximum packet loss rate required to achieve a target bandwidth.

Kernel instrumentation using web10g: Web10g [5] is basically the project of web100 developmed by supercomputing system center, pitsonburg. The web10g is the basic implementation of RFC4898 [6], The main goal of the Web10G to provide TCP instrumentation and /proc Application Binary Interface ABI which can be combined into a single kernel patch. Web10G provides the tools to diagnose to users and researchers. TCP hides the details which is critically important. To exposing this implementation of TCP as part of standard Linux. This implementation will expose TCP's hidden apparatus and allow any interested user to see how TCP actually works.

VI. IMPLIMENTATION AND RESULT DISCUSSION

Data Collection: We have used standard data set to test the CO-ABC based ANN development. As discuss the working of CO-ABC in the previous topics we have used matLab tool to test and implementation. The aim behind using different datasets for experimentation is to prove the consistency of the CO-ABC algorithm in different domains. The number of instances in these datasets range from hundreds to thousands, verifying the performance of CO-ABC algorithm on datasets with different sizes. Several experiments were performed in order to evaluate the accuracy of the ANN designed by means of the CO-ABC. The accuracy of the ANN was tested with seven pattern classification problems. Seven of were taken from UCI machine learning benchmark repository: iris plant, wine, breast cancer, heart disease, glass identification, image segmentation and wilt datasets. Based on above data set implementation we found the CO-ABC performing the best for exploration and exploitation issue [7] Base on the above experiment we have identify our basic data set with attribute 3, Instances 1016 and Classes 3. The experimental data collection table as known as Windowing_sets. The data is purely collected base on the connection we have received at our experimental servers with no restrictions. Based on collected data sets we calculated the actual require window size data set by modifying the data in excel sheet using real time equations for window size requirement base on RTT and MSS as unlimited. The data set is named as Windowing_sets_actual. We initialization ABC ANN with MNC value 800, limit 100, colony size 20, Transfer function as RBF, Greedy selection method and ObjFun are Weight, FF1 and FF2 with Lower and Upper bound value [-3,3], 18 experiments using each dataset were performed. Nine for the case of fitness function FF1, and nine for the case of fitness function FF2. For each experiment, each dataset was randomly divided into two sets: a training set and a testing set; for training & testing ratio is take 70, 30 % because from the stastical comparison of all data set we get high we get a high accuracy in 70, 30 % ratio compare to other ratio in case of training & testing for each dataset that are conclude table which we have not included over here because of document space restrictions.

Experimental Results of Errors with FF1 and FF2 function, Depending of the problem, the CO-ABC algorithm approaches the solution to the minimum error during the evolutionary learning process at different generations. For instance, for the case of the Windowing_sets_actuals dataset problem, we observed that by evolving FF1 (the one using MSE), the error tends to diminish faster and after a certain number of generations the error diminish slowly Fig 1. On the other hand, we also observed that, in some cases when FF2 is evolved, the error reaches the minimum error in a few number of epochs; none the less the error tends to diminish slowly Fig1.

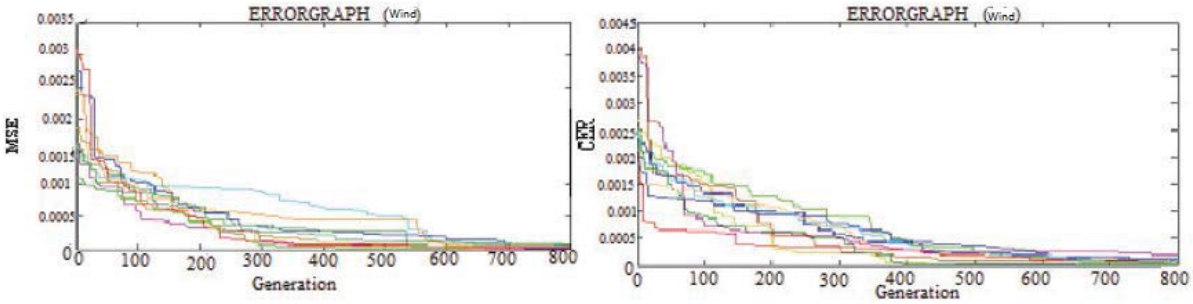

Figure 1: Evolution of FF1(MSE) & FF2(CER) functions for Windowing_sets_actual dataset

The heart of the algorithm are the functions FF1 and FF2 operations which lead to increase in accuracy from a smaller value to significantly high values. The average results of the FF1 and FF2 operations perform on all the stated datasets is shown the accuracy with F1,FF1,F2 and FF2 is 90.451,93.11,89.381,93.75 respectively also the Mean Square Error identify is 0.006, 0.0034, 0.003, 0.0023 respectively. The overall result shows the best performance of CO-ABC based Artificial Neural Network. We have also compare the ABC with CO-ABC for accuracy and MSE parameter and found CO-ABC better. We have not included all here for space limitation.

The figure 2, show the overall project work. Which include the Kernel mode, User mode and Application layer. The Linux kernel will support the TCP stack to fully open up. Kernel is the core of the operating system. Kernel mode mainly uses to restrict the unauthorized activities which may cause unstabilized state of basic operating system. While the user mode is at user space based on kernel platform to perform the basic programming based tasks. User mode and kernel mode here will interact with help of system calls. We have utilized the web10g kernel application packages interfaces to do the interaction with the kernel. Kernel Instrument set depicted in the data flow figure 2, above will be Initialize, send (one of various) messages, Returned data encapsulated in easily used data structures and

perhaps Monitor kernel events (say, of connection creation. It also respond to events by user-defined callbacks. List all connections owned by requesting uid, in the form of CID (connection ID; RFC 4898) which is combination of Local address, Local port, Remote address, Remote port. Read with mask here Request current values of all, or a subset of, Management Information base(MIB) vars for a specified CID and Returned data is an array of values, encapsulated in tcpe-data struct. One has the option of sending a mask with the read_conn request, specifying a subset of MIB vars. This limits the time spent holding the socket locked.

Over all algorithms: The proposed algorithm for Artificial Neural Network based connection and window parameter recommendation. This is per connection based recommendation by Artificial Neural Network. Before execution of this algorithms we make shore that the data collection data set Window_dataset_actual must be supply to CO-ABC based Artificial Neural network and train the NN as per the ratio of 70/30% as all we discussed in previous topics.

Algorithm 1: The Proposed Algorithm

1. *Design and Create local Type agent 'agent'*
2. *Create group 'group'*
3. *Identity address type 'add_Type'*
4. *While 'group' members available*
5. *Temp Disable kernel default buffer setting facility*
6. *Check local address type ipv4/ipv6*
7. *Read local address and port 'localadd', 'localport'*
8. *Read remote address and remote port 'RemAdd', 'RemPort'*

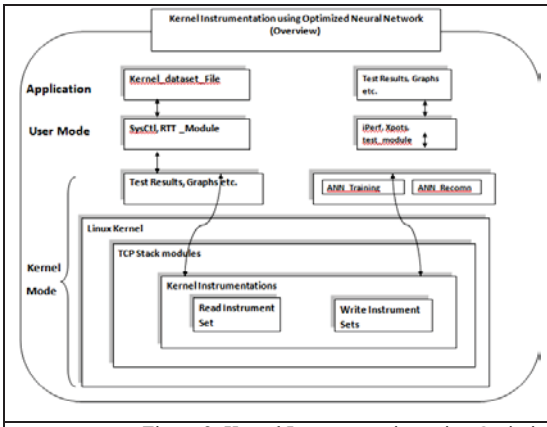

Figure 2: Kernel Instrumentation using Optimization NN

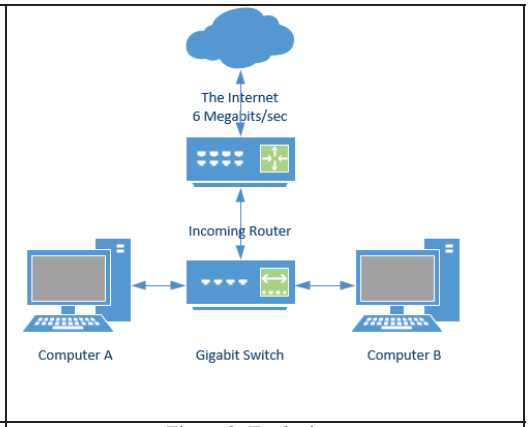

Figure 3: Testbed setup

9. *Generate connection ID 'CID'*
10. *Define snap 'snp'*
11. *If RemPort is not in reserved list*
12. *While Instruments in snap*
13. *If Instrument Con in "Established" state*
14. *Enable and Read ANN recommendation Module Ann_recmnd*
15. *Enable Window Scaling Accordingly for CID*
16. *Call Write_Kern_Paramtrs for CID*
17. *Disable SACK*
18. *End while instruments*
19. *Disable spoofin, sy-flood attack if require*
20. *Disable ICMP redirect if require*
21. *End while Group*
22. *Release memory if con closed*

Results discussion: We have already discuss the MSE and Fitness function to prove the CO-ABC based artificial neural network is better here we will discuss the testing and of the QoS parameter based on the recommendation of the Neural Network.

The project module starts up, to read /proc/web100 file and read per connection kernel instruments, convert IP and port address from kernel space to user space. It also converts the instruments value in normal data type for the display and modification. The connection will identify with the CID as displayed in algorithms. The per connection based CO-ABC based ANN will recommended the slected paremeters of kernel to support the TCP to fully openup. The parameters will be like rclWnd (Receiver Limiting Window) curr_cwnd (current congestion window), sending window, accuracy and Kern algorithms based RTT calculated bandwidth etc. We enable the scaling window size option for extension for the high-performance. We require to check the size of rclWnd, which should not exceed 12MB. If it is greater than 12MB and selective retransmission event occurs, then searching for interested packet takes a longer time than RTO (retransmission time out), which forces the slow start [8]. Slow start threshold (shThrsh) is used when switching from exponential increase to linear increase. The value for ssthresh for a given path is cached in the routine table If there is a retransmission on a connection to a given host, then all connections to that host for the next 10 minute will use a reduced ssthresh. Or, if previous connection to that host is practically good, then you might stay slow start in long time. We are using following sysctl (Configure Linux kernel parameters at runtime) parameter to disable the same. We are not listing some of the activities to save the document space.

Testing and Analysis: The network measurement tools available to application developers and system administrators are used to measure physical data-link bandwidth, round trip time, loss rate, router buffer sizes at each hop in the network, and measure end-to-end network bandwidth. The UNIX ping utility is used to transmit and receive ICMP Echo packets to a destination host to determine if the host is reachable, to measure round trip time (RTT), and to measure packet loss on the network path to the host. The RTT measurements made by ping can be used to estimate the ”pipe” capacity (capacity =BW * RTT) of the network between two hosts. Since the test load put on the network by ping consists of small periodic ICMP packets, the packet loss rate measured by ping is not very useful for estimating available TCP bandwidth using Equation discussed previously. The RTT (Smooth RTT) measurement, however, is useful for deriving the maximum packet loss rate necessary to support a desired TCP bandwidth. Traceroute [9], Ipef[10], TCPTrace[12], xPlot[13] are used to testing a specific function and meet requirement.

Figure 3. shows there are many senders and many receivers connected with high speed routers. All nodes having Realtek RTL8168/8111 PCI-E Gigabit Ethernet NIC and are connected with cat-6 cable and min 6Mbps of internet speed. The host to host testing with iperf, resulted in a difference of around 305Mbps of speed. For 16KB of window size we are having throughput of 132Mbps maximum while for 2MB of window size it is 438Mbps. This is shown in Figure 6.2 (next figure). We are transferring large avi file from computer-A to computer-B using FTP. We takes all the result for near about 2 minutes of transaction. We have use the combination of tcpDump, tcpTrace and xplots for per-connection throughput measurement with commands

Figure 6.3 shows the throughput using tcpDump, tcpTrace and xPlot per network connection port. Y-axis contain the window size in Kilo Bytes and the x-axis contain the time. The instantaneous throughput value by yellow dots are at around 150MB. The Figure 6.4, shows the window buffer after tuning Qos parameter. Y-axis contain the window size in Kilo Bytes and the x-axis contain the time. we can see maximum dots at 550MB. We can conclude that the throughput increase from 150-580Mbps. The Figure 6.5 show the instantaneous window size of tuned and un-tuned connection, the announce window is maximum 2896B for un-tuned connection, showing fixed window size for all the transaction because of the limited window buffer size

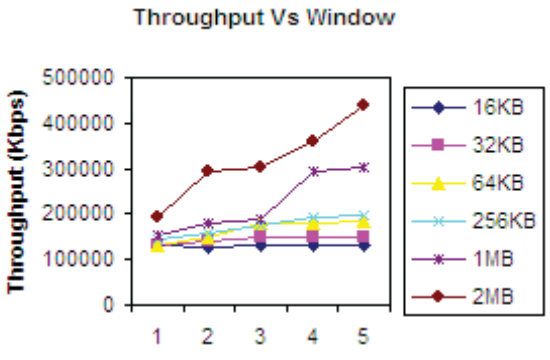

Figure 6.2: Throughput Vs Window size

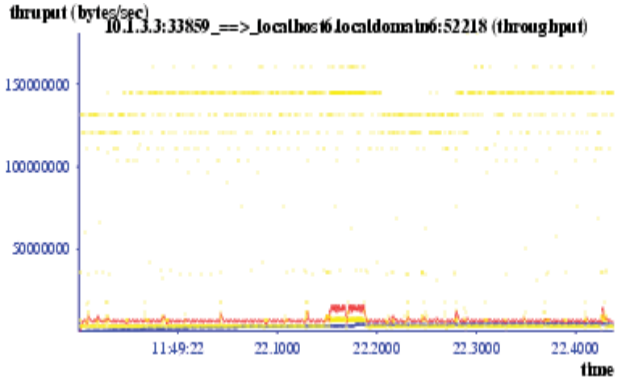

Figure 6.3: Throughput without ANN recomendation.

The Figure 6.5 also shows the announce window size which is different at every transaction. This is because at the receiving end the window size is allowed to take the recommendation of CO-ABC based ANN recommendation according to BDP. It conclude that for ANN here allow TCP to fully open-up.

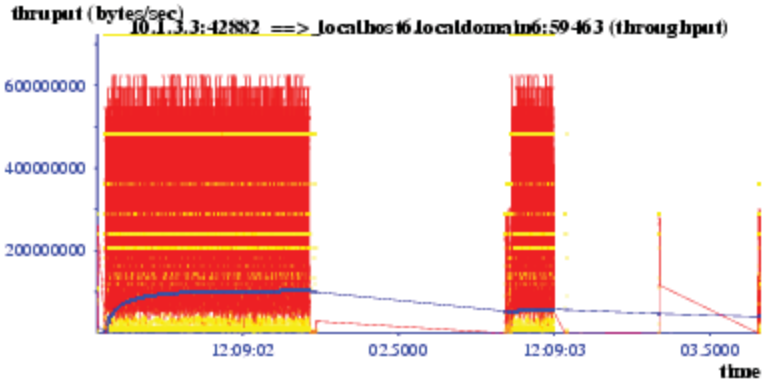

Figure 6.4: Throughput after ANN recommendation.

V. CONCLUSION

This paper demonstrates the kernel space and user space can be effectively used in combination with network instrumentation. The Kernel's TCP instrumentation using the proposed CO-ABC Nerual Network is a novel approach. We have identified requirements for QoS parameter instrumentation of TCP to achieve maximum throughput across all connections simultaneously within the resource limits of the sender. The technique to modify the TCP implementation of Linux kernel 3.11 to optimize the memory and bandwidth requirements of network connections results in greatly improved performance, a decrease in packet loss under bottleneck conditions, and greater control of buffer utilization by the end hosts. the basic research on the CO-ABC algorithm and the identification of need to modify the basic ABC algorithms compare to available modified version of ABC to overcome the exploitation and exploration problem in field of computer network.

REFERENCES

[1] W. Stevens, "Tcp slow start, congestion avoidance, fast retransmit, and fast recovery algorithms," RFC 2001, January 1997.

[2] VctorBerrocal-Plaza, "Artificial Bee Colony Algorithm Applied to WiMAX Network Planning Problem" IEEE- 2011

[3] Wei-Chang Yeh, Tsung-Jung Hsieh "Solving reliability redundancy allocation problems using an artificial bee colony algorithm", Computers & Operations Research, vol. 38, pp.1465-1473,2011.

[4] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the tcp congestion avoidance algorithm," Computer Communication Review, vol. 27, July 1997.

[5] www.web10g.org

[6] M. Mathis, J. Heffner, "TCP Extended Statistics MIB", RFC 4898, May 2007

[7] Nimbark Hitesh, Paresh Kotak and Nilesh Shah, (2012), "Intelligent computer networks: A Game Theoretic Approach to Compute the Traffic Equilibrium of Various Routing Schemes for multimedia applications in wireless network", IEEE Xplorer, Page: 407 – 411, Print ISBN: 978-1-4673-1538-8

[8] "Tcp tuning, www-didc.lbl.gov/tcp-tuning/linux.html,"

[9] V. Jacobson, "Traceroute: A tool for printing the route packets take to a network host,"

[10] Iperf tool, www.noc.ucf.edu/tools/iperf/,"

[11] V. Jacobson and C. Leres, "Tcpdump project,"

[12] Tcptrace, ohio university, "http://irg.cs.ohiou.edu/software/tcptrace/download.html,"

[13] Xplot project, " http://www.xplot.org,"