Exploring Round Trip Time fairness for Adaptive Layered Transmission Control Protocol

V.Kavidha
Department of Computer Science and Engineering, Dr Sivanthi Aditanar College of Engineering, Tiruchendur, Tamilnadu,INDIA
Email: kavidha.kavidha@yahoo.com

Dr. V.Sadasivam
Department of Computer Science and Engineering, Manonmaniam Sundarnar University,Abishekapatti, Tirunelveli,Tamilnadu, INDIA
Email: vs_raga@yahoo.com

---ABSTRACT---
High performance data transfer services is needed in long distance high-speed networks. In this paper Adaptive Layered Transmission Control Protocol (ALTCP) is proposed, which is used for making more scalability in high-speed networks. ALTCP is a simple adaptive layering technique for making the Additive Increase Multiplicative Decrease (AIMD) algorithms used by TCP more efficient in probing for the available link bandwidth. ALTCP uses a three-dimensional congestion control framework. First the macroscopic control is employed to layer quickly and made efficient by using available link bandwidth, second microscopic control is used for extends the existing AIMD algorithm of TCP to determine the per acknowledgement behavior. Third the intermediate control is employed for decoupling the aggregate throughout from the number of opened TCP flows in parallel. In this research paper ALTCP protocol is designed and analyzed based on ns-2 simulations. The results show that ALTCP has faster magnitude than TCP in utilizing high bandwidth links. ALTCP has better TCP friendliness and Round Trip Time (RTT) fairness compared with high-speed protocols namely High-speed TCP and Scalable TCP.

Key words - AIMD, ALTCP, Congestion Control, High speed networks, RTT fairness

Date of Submission: 24, December 2009 Date of Acceptance: 19, April 2010

1. INTRODUCTION

The performance of the Internet is determined not only by the network and hardware technologies that underlie it, but also by the software protocols that governs its use. In particular, the TCP transport protocol is responsible for carrying the great majority of traffic in the current internet, including web traffic, email, file transfers, music and video downloads. TCP provides two main functions. First, it provides functionality to detect and retransmit packets lost during a transfer thereby providing a reliable transport service to higher layer applications. Second, it enforces congestion control that is it seeks to match the rate at which packets are injected into the network to the available network capacity. The TCP congestion control algorithm has been remarkably successful in making the current Internet function efficiently. However, in recent years it has become clear that it can perform very poorly in networks with high Bandwidth Delay Product (BDP) paths. The problem stems from the fact that the standard TCP, AIMD congestion control algorithm increases the congestion window too slowly. Congestion control is an important component of a transport protocol in a packet-switched shared network.

This paper proposes Adaptive Layered Transmission Control Protocol (ALTCP), with a set of modifications to the congestion window response of TCP to make it more scalable in high-speed networks. ALTCP modifies the TCP flow by using the concept of virtual layers, such that the convergence properties and Round Trip Time fairness behavior is maintained similar to that of TCP. This paper provides the perception design for the ALTCP protocol modifications and evaluation results based on ns-2 simulations and Linux implementation. This paper also proposes an evaluation method of Round Trip Time fairness for ALTCP with high-speed protocols namely High-speed TCP and Scalable TCP.

2. RELATED WORK

Kelly had developed congestion control algorithm for widely used transport protocol which is responsible for detecting and reacting to overloads in the Internet and has been the key to the Internet's operational success [1]. Jim Martin had developed many protocols which differ mainly in their choices of window adjustment algorithms, particularly used in the functions growth phase of the congestion window [2]. The choices of growth functions are diverse from exponential to some polynomial functions. While a number of proposals have been made to modify the TCP congestion control algorithm, all of these are still experimental and pending evaluation as they change the congestion control in new and significant ways and their effects on the network are not well understood. In fact, the basic properties of networks employing these algorithms may be very different to networks of standard TCP flows. The TCP congestion control algorithm has been remarkably successful in making the current Internet function efficiently. However, in recent years it has become clear that it can perform very poorly in networks with high Bandwidth Delay Product paths [3]. The problem stems from the fact that the standard TCP, AIMD congestion control algorithm increases the congestion window too slowly.

Kunniyur and Srinath had developed a framework for evaluating congestion control algorithms [4]. The framework includes a number of metrics such as throughput, packet loss rates, delays, and fairness as well as a range of network environments. The framework illustrate the need for realistic performance evaluations of new congestion control algorithms and emphasize the motivation for this work and existing evaluation work that briefly review below. Sumitha Bhandarkar had developed Layered TCP scheme which modifies the congestion response function of TCP at the sender-side and requires no additional support from the network infrastructure or the receivers [5]. The key contribution of Layered TCP is that it emulates multiple virtual flows that adapt to the dynamic network conditions by using a simple layering technique. Layered TCP in contrast to this earlier body of work considers window adaptation at each layer in addition to adding/dropping layers, and considers fairness issues.

Tom Kelly had developed Scalable TCP which offers a robust mechanism to improve performance in high-speed wide area networks using traditional TCP receivers [6]. Scalable TCP uses multiplicative increase/multiplicative decrease response, to ensure that the congestion window can be doubled in a fixed number of RTT [7]. David had developed FAST TCP relies on the delay based bandwidth estimation of the TCP Vegas and is optimized for Gbps links [8]. FAST TCP addresses the four main problems of TCP Reno in networks with high capacities and large latencies. It has a log utility function and achieves weighted proportional fairness. Li had developed many studies to improve the performance of TCP protocol [9]. Mohamed and Robert had developed new reno TCP designed to be incrementally deployable and behave identically to traditional TCP stacks when small windows are sufficient [10]. Shorten had developed log utility function for achieving weighted proportional fairness [11].

3. ADAPTIVE LAYERED TCP

In Macroscopic design, ALTCP employs an RTT Compensation factor K_R based on the RTT observed by the flow. The RTT Compensation technique modifies the congestion window update algorithm, on the receipt of an acknowledgment, for ALTCP.RTT Compensation uses a scaling factor K_R to the basics ALTCP window increase function. On a successful receipt of one window of acknowledgements, ALTCP will increase its congestion window by K_R*K packets, instead of K packets. For this design equation is:

$$\frac{W_{R1}}{(K_{R1}*K)} > \frac{W_{R2}}{(K_{R2} * K_2)} \qquad (1)$$

Where $W_{R1,}$ W_{R2} are the window reductions for each flow upon a packet loss and $K_{R1,}$ K_{R2} is the scaling factor for each flow.

In Microscopic design, two modes of operations for ALTCP are defined. In the steady state, the protocol need not be aggressive. Therefore set $K_R = 1$ or turn it off. This will reduce the drop rates in steady state. In the transient state, a flow is probing for the available bandwidth and needs to be more aggressive. Therefore, we turn on K_R

$$K_R = \begin{cases} 1, \text{ during steady state (off)} \\ 0.5(RTT)\ 1/3, \text{ during transient state (on)} \end{cases} \qquad (2)$$

In intermediate behavior the idea behind Adaptive TCP is to quickly increase the strength k to effectively utilize available bandwidth when there is no competition, and enter non-aggressive mode by setting the strength of Adaptive TCP to $k = 1$ to ensure TCP-friendliness whenever it detects competition with other flows. Note that Adaptive TCP is identical to standard TCP when its strength $k = 1$. In addition, Adaptive TCP sets $k = 1$ during periods of congestion to reduce packet loss regardless of the result of competition detection. Here, by congestion mean that the network queuing delay estimated by Adaptive TCP is larger than a reference queuing delay level [12]. This work uses the same value for this as the threshold queuing delay level used to trigger the competition detector. Adaptive TCP uses the queuing delay increase as an estimator for congestion and competition. It does so by keeping track of the minimum and maximum of the smoothed Round Trip Time 5(*srtt* min and *srtt* max, respectively) and triggers detection when the estimated queue build-up (*srtt* − *srtt* min) exceeds some threshold [13].

4. PROTOCOL IMPLEMENTATION

The pseudo-code of the ALTCP is presented below. A packet drop event is characterized by the receipt of triple-duplicate acknowledgements from the receiver.

Volume:01, Issue: 06, Pages: 353-358 (2010)

The following parameters are used,

ALTCP Algorithm:

Current_ layer : layer at which packet drop event occurred.
Last_ layer : layer at which last packet drop event occurred.
Second_ last_ layer : layer at which second last packet drop event occurred.
K : current operating layer.
W_K :window corresponding to the layer, K
stored K_R : stored value of K_R which is calculated at the start of the flow and updated

Whenever minimum *RTT* changes.

Initialization:

Second_ last_ layer = *last_ layer* =*current_ layer* = 1;
stored K_R= 0.5(*RTT*)1/3;

On receiving 3 duplicate acknowledgments, decrease congestion window:

Second_ last_ layer = *last_ layer*;
Last_ layer = *current_ layer*;
Current_ layer = *K*;
if (*second_ last_ layer* ≥ *last_ layer* && *last_ layer* ≥ *current_ layer*) **then**
K_R= 1;
else
K_R= stored K_R;
$cwnd = (1 - \beta) * cwnd$;
while $cwnd < W_K$ do
$K = K - 1$;
end while
end if

On receipt of an acknowledgment, increase congestion window:

$cwnd = cwnd + (K_R * K)/cwnd$;
While $cwnd > W_{K+1}$ **do**
//window crosses the current layer boundary. Increase number of layers
K + +;
if *K* > current layer **then**
// layer crosses the layer at which last drop occurred.
K_R= stored K_R;
end if
end while

4.1 Fairness of ALTCP

Fairness and efficiency are the center for most of the researchers conducted to evaluate the performance of high-speed TCP protocols. The fairness is measured by sharing the bottleneck bandwidth among competing flows that have different RTTs. There are several notions of “RTT fairness”. One notion is to achieve the equal bandwidth sharing where the two competing flows may share the same bottleneck bandwidth even if they have different RTTs. This property may not be always desirable because long RTT flows tend to use more resources that short RTT flows since they are likely to travel through more routers over a long path. Another notion is to have bandwidth shares inversely proportional to the RTT fairness.

This proportional fairness makes more sense in terms of the overall end to end resource usage. Although there is no commonly accepted notion of RTT- fairness, it is clear that the bandwidth share ratio should be within some reasonable bound so that no flows are being starved because they travel a longer distance. Note that RTT-fairness is highly correlated with the amount of randomness in packet losses or in other words, the amount of loss synchronization. In more random environments, protocols tend to have better RTT-fairness. In states A and D, both the flows have the same value of K_R . The convergence analysis will hold true and the two flows will converge. In the state C, K_R is turned off for the higher flow and turned on for the smaller flow[14]. After a packet drop, the convergence equation (1) of ALTCP can be written as:

$$W_{R1} / K_R * K_1 > W_{R2} / K_{R2} * K_2 \Rightarrow$$

$$W_{R1}/K_1 > W_{R2} / K_{R2} * K_2 \qquad (3)$$

By ALTCP design, the above equation holds true and the two flows will convergence.

4.2 Fairness of Adaptive TCP

Adaptive TCP flow in steady state can be modeled as:

$$T = \frac{\text{data sent per congestion epoch}}{\text{Time per congestion epoch}}$$

$$= \frac{W^2(4k-1/8k^2)}{W/2k\,(RTT)} \qquad (4)$$

$$T = \frac{\sqrt{(4knee\_k-1)/2}}{RTT\sqrt{p}} \quad \text{[packets per second]}$$

Where *k* is the strength of adaptive TCP and knee is the delay gradient However, in contrast to the fixed strength *k* in TCP/*k*, the strength *k* of Adaptive TCP is dynamically decided by knee *k* [15]. Hence, replace *k* with $\sqrt{knee\ k}$ and obtain the following equation:

$$T = \sqrt{(4\sqrt{knee\ k} - 1/2)} / \text{RTT}\sqrt{p} \qquad (5)$$

5. SIMULATION

ALTCP is designed using experiments conducted on ns-2 network simulator. All simulations are conducted using a dumb-bell network topology as shown in Figure-1. One common bottleneck link connects n sources to n corresponding receivers. Unless otherwise specified, the bottleneck link capacity is set to 1Gbps with a delay of 40ms. Links that connect senders and receivers to the routers are set to a bandwidth of 2.4Gbps and a delay of 10ms. Thus, end-to-end RTT for each flow is set to 120ms, unless specified. The default queue size at the routers is set to be equal to the product of bottleneck link bandwidth and delay. Drop-tail queue management scheme is used at the routers. The protocol is implemented by introducing a new window option in the basic TCP code in the file tcp.cc in ns-2. All the simulations use TCP/Sack1 agent for the sender and TCPSink/Sack1 agent for the receiver. Unmodified TCP/Sack1 is used for the TCP simulations. FTP traffic is used between the senders and receivers. All the readings are taken for 1000 seconds and data for initial 300 seconds is discarded, to ensure that steady state is reached.

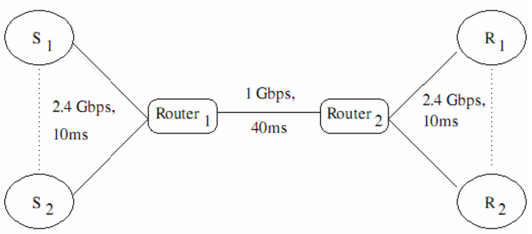

Figure-1 Simulation Topology

For the simulation of Adaptive TCP a simple dumbbell topology is used as shown in Figure 1 with bottleneck link bandwidth of 622Mbps and 50ms RTT propagation delay between senders and receivers. The bottleneck switch SW 1 uses FIFO/Drop-Tail scheduling with 20% of the bottleneck bandwidth-delay product. TCP-SACK is again used for the competing standard TCP, and TCP receivers use delayed ACK. TCP initial congestion window size is set to two packets instead of one to remedy round-tip time measurement errors by the use of delayed ACKs when connections initiate. FTP is used for application to generate continuously back-logged, long-term flows.

6. RESULTS

Since ALTCP uses adaptive layering, it is capable of increasing its window size to the optimal value much faster than TCP. Also, when a packet loss occurs, the window reduction of ALTCP is not as drastic as TCP. As a result the window adaptation of ALTCP is much more efficient in utilizing the link bandwidth in high-speed networks. Figure-2 shows congestion window of ALTCP in comparison with that of TCP, when the network consists of only one flow. The congestion window of ALTCP reaches the optimal value several orders of magnitude faster than the TCP flow. Figure-3 shows Adaptive TCP share the bandwidth fairly during the overlapping 100 seconds.

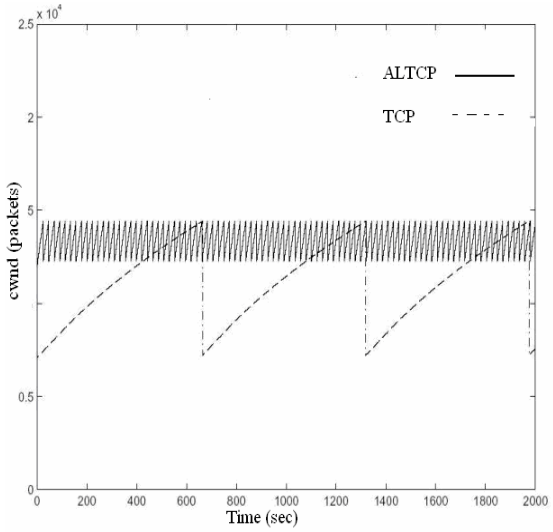

Figure-2 Congestion window behavior of ALTCP

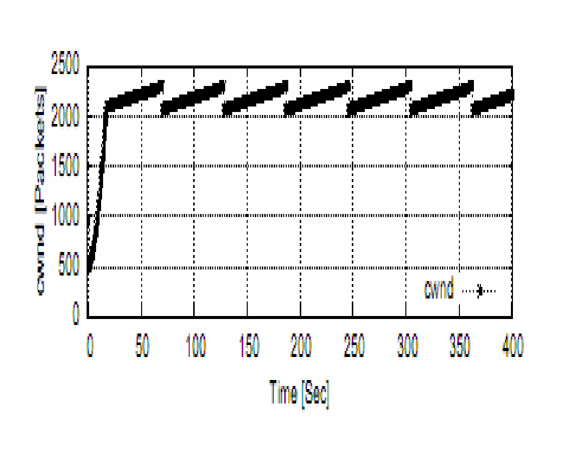

Figure-3 Congestion window behavior of Adaptive TCP

6.1 Fairness with same RTT:

Figure-4 plots the ratio of measured throughputs for two flows with the same propagation delay sharing a common bottleneck link as the path propagation delay is varied. Tests are of 10 minutes duration. Result is shown for a bottleneck link bandwidth of 250Mb/s, roughly corresponding to low and high-speed network conditions. The result shown is with no web traffic, but similar behavior is observed when web traffic is present. It can be seen that this basic test reveals some striking behavior. Under these conditions, the standard TCP congestion control algorithm consistently ensures that each flow achieves the same (to within less than 5%) average throughput. However, the measurements shown indicate that many of the proposed protocols exhibit substantial unfairness under the same conditions, while both FAST-TCP and Scalable-TCP display very large variations in fairness. ALTCP exhibiting greater levels of RTT fairness than FAST –TCP and Scalable – TCP.

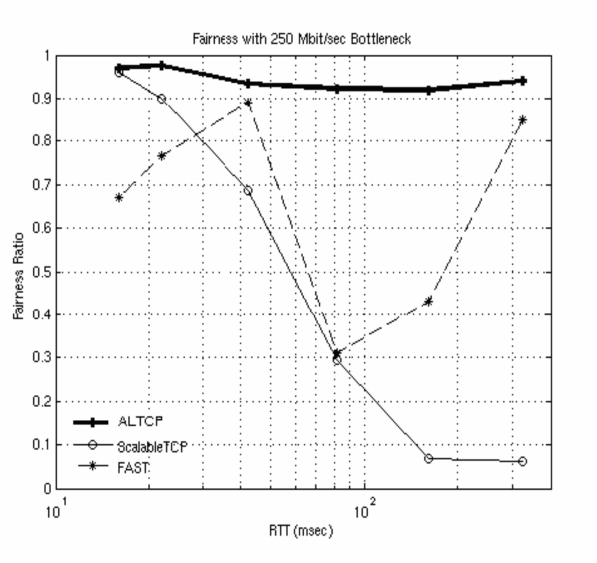

Figure-4 RTT fairness for ALTCP

6.2 RTT fairness of A-TCP

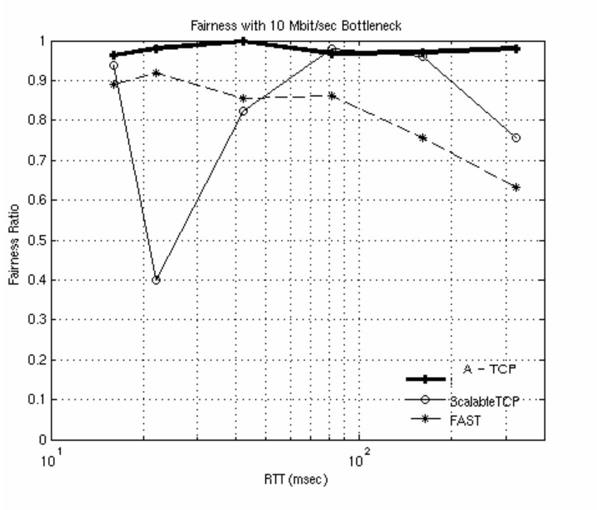

Figure-5 RTT fairness for A-TCP

Figure-5 plots the ratio of measured throughputs for two flows with the same propagation delay sharing a common bottleneck link as the path propagation delay is varied. Tests are of 10 minutes duration. Result is shown for a bottleneck link bandwidth of 10 Mb/s, roughly corresponding to low and high-speed network conditions. The result is shown with no web traffic, but similar behavior is observed when web traffic is present. It can be seen that this basic test reveals some striking behavior. Under these conditions, the standard TCP congestion control algorithm consistently ensures that each flow achieves the same (to within less than 5%) average throughput. However, the measurements shown indicate that many of the proposed protocols exhibit substantial unfairness under the same conditions, while both FAST-TCP and Scalable-TCP display very large variations in fairness. Adaptive TCP exhibiting greater levels of RTT fairness than FAST –TCP and Scalable – TCP.

7. CONCLUSION

In this work, suggests several congestion control schemes for single and parallel TCP flows. The collaborative congestion control scheme has shown the usefulness and benefit of collaboration among parallel TCP flows by sharing dynamic congestion information. ALTCP a new protocol is designed for improving the performance of window-based schemes in networks characterized by long-delay and high RTTs. This paper had provided the ground work for a new protocol set termed ALTCP. The protocol uses a set of RTT Compensation techniques to tune the performance of high-speed protocols in high RTT networks. Adaptive TCP, another protocol is designed to decouple the number of flows from the aggregated aggressiveness of the group of flows in a self managing fashion. Adaptive TCP which adequately adjusts the group strength k of parallel TCP flows to achieve high utilization of available bandwidth while maintaining TCP-friendliness against single TCP flows. The results show that ALTCP and Adaptive TCP have faster magnitude than TCP in utilizing high bandwidth links. ALTCP and Adaptive TCP have better TCP friendliness and RTT fairness compared with high-speed protocols namely High-speed TCP and Scalable TCP.

REFERENCES

[1] F P Kelly, Fairness and stability of end-to-end congestion control, *European Journal of Control*, 9(2), 159–176.

[2] Jim Martin, Arne Nilson, and Injong Rhee, Delay-based Congestion avoidance for TCP, *IEEE/ACM Transactions on Networking*, *11*(3), 2003, 356-369.

[3] H Sivakumar, R L Grossman, M Mazzucco, Y Pan, Q Zhang, Simple Available Bandwidth Utilization Library for High-Speed Wide Area Networks, *Journal of Supercomputing*, *34*(3), 2005, 231-242.

[4] S Kunniyur, R Shrikanth, End-to-end congestion control schemes: utility functions, random losses and ECN marks, *IEEE/ACM Transactions on Networking*, *11*(5), 2003, 689-702.

[5] Sumitha Bhandarkar, A L Narasimha Reddy and Saurab Jain, LTCP: **I**mproving the performance of TCP in high-speed networks, *ACM SIGCOMM Computer Communication Review* *36*(1), 2006, 41 – 50.

[6] Tom Kelly, Scalable TCP: Improving Performance in High-Speed Wide Area Networks, *ACM SIGCOMM Computer Communications Review*, *33*(2), 2003, 83-91.

[7] S Low, F Paganini, J Wang, and J Doyle, Linear Stability of TCP/RED and a Scalable Control, *Computer Networks Journal*, *43*(5), 2003, 633-647.

[8] X W David, Cheng Jin, H Low Steven and Sanjay Hedge, FAST TCP: Motivation, Architecture, Algorithms, Performance, *IEEE/ACM Transactions on Networking*, *14*(6), 2006, 1246-1259.

[9] Li Yee-Ting,D J Leith and R Shorten, Experimental Evaluation of High-Speed Congestion Control protocols, *IEEE/ACM Transactions on Networking*, *15*(5),2007, 1109-1122.

[10] Mohamed Tekala and Robert Szabo, Modeling scalable tcp friendliness to new reno tcp, *International Journal of Computer Science and Network Security*, *7*(3), 2007, 89-96.

[11] R Shorten, F Wirth, and D Leith, A positive systems model of tcp-like congestion control: asymptotic results, *IEEE/ACM Transactions on Networking*, *14*(3), 2006, 616–629.

[12] R Shorten., D Leith, On AIMD Congestion control, queue provisioning and network efficiency, *IEEE/ACM Transactions on Networking*, 15(4), 2007, 866-877.

[13]K S Reddy and L C Reddy, A survey on congestion control mechanisms in high speed networks, *International Journal of Computer Science and Network Security*, *8(*1), 2008, 208-212.

[14] Yunhong Gu and L Robert Grossman, UDT: UDP-based Data Transfer for High-Speed Wide Area Networks, *Computer Networks Journal*, *51*(7), 2007, 1777-1799.

[15] S Low, F Paganini, J Wang, and J Doyle, Linear Stability of TCP/RED and a Scalable Control, *Computer Networks Journal*, *43*(5), 2003, 633-647.

Authors Biography

Mrs.V.Kavidha has obtained her B.E., M.E., in 1999 and 2004 respectively. She is working as Senior Lecturer in Dr.Sivanthi Aditanar College of Engineering, Tiruchendur. She has 11 years of teaching experience. She is a Life member of Indian Society for Technical Education and Associate member in Institution of Engineers. Her interesting research area is "Networking". She has presented three papers in National Conference. She has published one paper in National Journal.

Dr.V.Sadasivam has obtained his B.E in Electrical Engineering and M.Sc (Engg) in Power System in 1973 and 1975 respectively. He Completed his Ph.D in Computer Science and Engineering in 1993. From 1995 onwards, he is working as Professor and Head of the Department of Computer Science and Engineering at Manonmanaim Sundarnar University, Tirunelveli. He has published 20 papers in International and National Journals and more than 60 papers in National and International Conferences. His areas of Interest are Artificial Intelligence, Image Processing and Fuzzy logic.