International Journal of Scientific Engineering and Technology Research

ISSN 2319-8885
Vol.03,Issue.48
December-2014,
Pages:9788-9793

www.ijsetr.com

A Fuzzy Logic Control in Distributed Traffic Management for Efficient Networks

KISHORE TEPPALA[1], K.RAVI KUMAR[2]

[1]PG Scholar, Dept of Software Engineering, Kakinada Institute of Engineering & Technology, JNTUK, AP, India,
E-mail: kishore.yfc@gmail.com.

[2]Associate Professor, Dept of CSE, Kakinada Institute of Engineering & Technology, JNTUK, AP, India,
E-mail: ravikumark@kietgroup.com.

Abstract: Wireless sensor networks (WSANs) are emerging rapidly as a new generation of sensor networks. Despite intensive research in wireless sensor networks (WSNs), limited work has been found in the open literature in the field of WSANs. In particular, quality-of-service (QoS) management in WSANs remains an important issue yet to be investigated. As an attempt in this direction, this paper develops a fuzzy logic control based QoS management (FLC-QM) scheme for WSANs with constrained resources and in dynamic and unpredictable environments. Taking advantage of the feedback control technology, this scheme deals with the impact of unpredictable changes in traffic load on the QoS of WSANs. This paper we propose a fuzzy based congestion control approach to address the congestion control problem. The performance of the proposed controlled system is evaluated via simulation. In view of the fast-growing Internet traffic, this paper propose a distributed traffic management framework, in which routers are deployed with intelligent data rate controllers to tackle the traffic mass. Simulation results and comparisons have verified the effectiveness and showed that our new traffic management scheme can achieve better performances than the existing protocols that rely on the estimation of network parameters.

Keywords: Congestion Control, Fuzzy Logic Control, Quality Of Service, Max-Min Fairness, Robustness, Traffic Management.

I. INTRODUCTION

WSNs are typically used for information gathering in applications like habitat monitoring, military surveillance, agriculture and environmental sensing, and health monitoring. The primary functionality of a WSN is to sense and monitor the state of the physical world. In most cases, they are unable to affect the physical environment. However, in many applications, observing the state of the physical system is not sufficient, it is also expected to respond to the sensed events/data by performing corresponding actions on the system. This stimulates the emergence of wireless sensor/actuator networks (WSANs) [5,6]. Featuring coexistence of sensors and actuators, WSANs enable the application systems to sense, interact, and change the physical world. They can be deployed in lots of applications such as disaster relief, planet exploration, intelligent building, home automation, industrial control, smart spaces, pervasive computing systems, and cyber-physical systems. Real-world WSAN applications have their requirements on the quality of service (QoS). For instance, in a fire handling system built upon a WSAN, sensors need to report the occurrence of a fire to actuators in a timely and reliable fashion; then, the actuators equipped with water sprinklers will react by a certain deadline so that the situation will not become uncontrollable.

Both delay in transmitting data from sensors to actuators and packet loss occurring during the course of transmission may potentially deteriorate control performance of the system, and may not be allowed in some situations where the systems are safety-critical. In a smart home, although there is no hard real-time constraint, actuators should turn on the lights in a timely fashion once receiving a report from sensors when someone enters or will enter a room where all lights are off; people would get unsatisfied if kept staying in dark for a long time waiting for lighting. In practice, QoS requirements differ from one application to another; however, they can be specified in terms of reliability, timeliness, robustness, trustworthiness, and adaptability, among others. Some QoS metrics may be used to measure the degree of satisfaction of these services. Technically, QoS can usually be characterized by, e.g., delay and jitter, packet loss, deadline miss ratio, and/or network utilization (or throughput) in the context of WSANs. As an alternative, a class of explicit congestion control protocols has been proposed to signal network traffic level more precisely by using multiple bits. Examples are the XCP [6], RCP [11], JetMax [12] and MaxNet [13]. These protocols have their controllers reside in routers and directly feed link information back to sources so that the link bandwidth could be efficiently utilized with good scalability and stability in high BDP networks.

Specifically, JetMax and MaxNet signal network congestion by providing the required fair rate or the maximum link price, and then the final sending rate is decided by sources according to some demand functions or utility functions. XCP feeds back the required increment or decrement of the sending rate, while RCP directly signals sources with the admissible sending rate according to which sources pace their throughput. The advantages of these router-assisted protocols are that 1) they can explicitly signal link traffic levels without maintaining per-flow state, and 2) the sources can converge their sending rates to some social optimum and achieve a certain optimization objective [12]. However, most of these explicit congestion control protocols have to estimate the bottleneck bandwidth in order to compute the allowed source sending rate or link price. Recent studies show that misestimation of link bandwidth (e.g., in link sharing networks or wireless networks) may easily occur and can cause significant fairness and stability problems [14], [15]. There are some latest protocols on wireless applications such as QFCP (Quick Flow Control Protocol) [16] and the three protocols called Blind, ErrorS and MAC [17]. They have improved on the estimation error while having high link utilization and fair throughput. However, they still have the fundamental problem of inaccurate estimation resulting in performance degradation.

In addition, their bandwidth probing speed may be too slow when the bandwidth jumps a lot. Also, they cannot keep the queue size stable due to oscillations, which in turn affect the stability of their sending rates. There are some explicit protocols that appear to compute the sending rates based solely on the queue size, but in fact they still need to estimate the number of active flows in a router, and this consumes CPU and memory resources. Examples are the rate-based controllers [18]–[20] for packet switching networks and the ER (Explicit Rate) allocation algorithm [21] for ATM (Asynchronous Transfer Mode) networks. For the API-RCP controller [19], both the original method (a truncated network model) and the improved method [22] face a memory problem when dealing with many flows (that numbers in millions) arriving to a core router every hour [23]. In some other controllers (e.g., [21]), the TBO (Target Buffer Occupancy) is designed to be as high as 3 times of the BDP, which can cause large queueing delay and thus degrading network performance, and this becomes even worse in the high-speed networks.

Historically, the ER allocation algorithms in ATM networks also share the same problems (e.g., [21], [24]) because they need to evaluate the link bandwidth and/or the numbers of active VCs (Virtual Circuits). Some others (e.g., [25]) adjust the source sending rates in binary-feedback switches or explicit feedback switches according to a few queue thresholds, which may cause unfairness as well as high cell loss rate [2], [26]. From the perspective of network and service management, the aforementioned congestion control approaches have QoS (Quality of Service) problems in that they cannot guarantee a certain level of performance under some situations due to design drawbacks. There are many different approaches to improve QoS. For example, admission control, as a network traffic management approach, can guarantee QoS by checking the availability of network bandwidth before establishing a connection, e.g., [27]–[29]. Service priority as another approach can be used to improve QoS by providing different service priorities to different users, e.g., [30]–[32]. Pricing or routing policies are also found to address QoS problems e.g., [33]–[35]. However, they are outside the scope of this paper that focuses on congestion control as an approach to address the QoS management problem.

FLC (Fuzzy Logic Control) [36] has been considered for IC (Intelligence Control). It is a methodology used to design robust systems that can contend with the common adverse synthesizing factors such as system nonlinearity, parameter uncertainty, measurement and modeling imprecision [37]. In addition, fuzzy logic theory provides a convenient controller design approach based on expert knowledge which is close to human decision making, and readily helps engineers to model a complicated non-linear system. In fact, fuzzy logic control has been widely applied in industrial process control and showed extraordinary and mature control performance in accuracy, transient response, robustness and stability [38],[39]. FLC has found its applications to network congestion control since 1990. In early stage, it was used to do rate control in ATM network, e.g., [40], [41], to guarantee the QoS. These control algorithms are explicit in nature, and they depend on absolute queue length (the maximum buffer size) instead of the TBO to adjust the allowed sending rate. Nevertheless, these early designs have various shortcomings including cell loss (even though cell loss is used as a congestion signal to compute the rate factor, e.g., [42]), queue size fluctuations, poor network latency, stability and low utilization.

Later, FLC was used in RED (Random Early Detection) algorithm in TCP/IP networks, e.g., [43], [44], to reduce packet loss rate and improve utilization. However, they are still providing implicit or imprecise congestion signaling, and therefore cannot overcome the throughput fluctuations and conservative behavior of TCP sources. In light of the above review of different protocols and their shortcomings, we would like to design a distributed traffic management scheme for the current IP (Internet Protocol) networks (and the next generation networks where applicable), in which routers are deployed with explicit ratebased congestion controllers. We would like to integrate the merits of the existing protocols to improve the current explicit traffic congestion control protocols (like XCP, RCP, APIRCP and their enhancements) and form a proactive scheme based on some prudent design ideas such that the performance problems and excessive resource consumption in routers due to estimating the network parameters could be

overcome. In this respect, a fuzzy logic controller is quite attractive because of its capability and designing convenience as discussed above.

Specifically, the objectives of this paper are: 1) to design a new rate-based explicit congestion controller based on FLC to avoid estimating link parameters such as link bandwidth, the number of flows, packet loss and network latency, while remaining stable and robust to network dynamics (Hence, we make this controller "intelligent"); 2) to provide maxmin fairness to achieve an effective bandwidth allocation and utilization; 3) to generate relatively smooth source throughput, maintain a reasonable network delay and achieve stable jitter performance by controlling the queue size; 4) to demonstrate our controller has a better QoS performance through case study. To achieve the above objectives, our new scheme pays attention to the following methodologies as well as the merits of the existing protocols. Firstly, in order to keep the implementation simple, like TCP, the new controller treats the network as a black box in the sense that queue size is the only parameter it relies on to adjust the source sending rate.

The adoption of queue size as the unique congestion signal is inspired by the design experience of some previous AQM controllers (e.g., RED and API-RCP) in that queue size can be accurately measured and is able to effectively signal the onset of network congestion as shown in Fig.1. Secondly, the controller retains the merits of the existing rate controllers such as XCP and RCP by providing explicit multi-bit congestion information without having to keep per-flow state information. Thirdly, we rely on the fuzzy logic theory to design our controller to form a traffic management procedure. Finally, we will employ OPNET modeler to verify the effectiveness and superiority of our scheme. The contributions of our work lie in: 1) using fuzzy logic theory to design an explicit rate-based traffic management scheme (called the IntelRate controller) for the high-speed IP networks as shown in Fig.2; 2) the

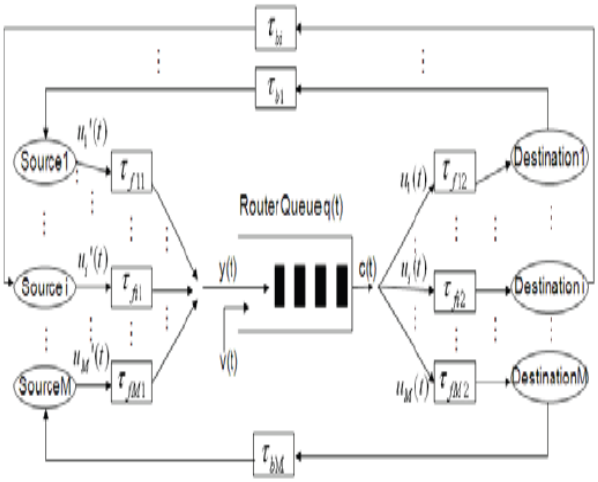


Fig. 1. System model of an AQM router.

application of such a fuzzy logic controller using less performance parameters while providing better performances than the existing explicit traffic control protocols; 3) the design of a Fuzzy Smoother mechanism that can generate relatively smooth flow throughput; 4) the capability of our algorithm to provide max-min fairness even under large network dynamics that usually render many existing controllers unstable.

The rest of the paper is organized as follows. After a description of network model and assumptions in Section II, Section III introduces the design rationale and the controller implementation procedure.

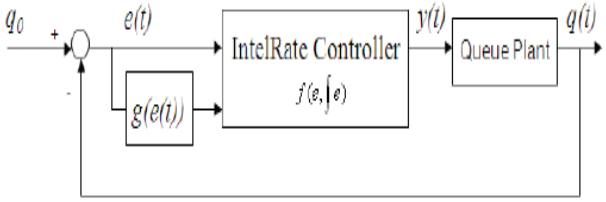


Fig. 2. The IntelRate closed-loop control system.

II. PERFORMANCE EVALUATION

The capability of the IntelRate controller is demonstrated by performance evaluations through a series of experiments. We will first describe the simulated network and performance measures of a case study in Section A. Section B demonstrates the system robustness upon large network changes. Queueing jitter control and the effect of short-lived traffic will be discussed in Sections C and D. Section E evaluates the bandwidth utilization and packet loss rate of the IntelRate controller. Finally, Section F discusses the choices of some design parameters. For the following evaluation, we choose N = 9, m = 8 and the delay of TBO≤10ms, while B = 10∗q0 and d = 50ms using the design principles for the IntelRate controller discussed in Section III. As noted there, some of these "optimum/good" values are the results after many experiments on different combinations. Due to space limitation, they are not presented.

A. Simulated Network

The single bottleneck network in Fig. 6 is used to investigate the controller behavior of the most congested router. We choose Router 1 as the only bottleneck in the network, whereas Router 2 is configured to have sufficiently high service rate and big buffer B so that congestion never happens there. The numbers in Fig. 3 are the IDs of the subnets/groups attached to each router. Their configuration is summarized in Table II, in which there are M = 11 subnet pairs, which form the source-destination data flows in the network, and they run various Internet applications such as the long-lived ftp, short-lived http, or the unresponsive UDP-like flows (also called uncontrolled ftp flows [19]). Since the link bandwidth we want to simulate have a magnitude of Giga bits per second, we need

to use 20 flows in each subnet to generate enough traffic to produce congestion. All flows within each group have the same RTPD and behavior, but different from the flows of other groups. The RTPD includes the forward path propagation delay and the feedback propagation delay, but does not include the queueing delay, which may vary according to our settings of TBO size in the experiments. The reverse traffic is generated by the destinations when they piggyback the ACK information back to the sources.

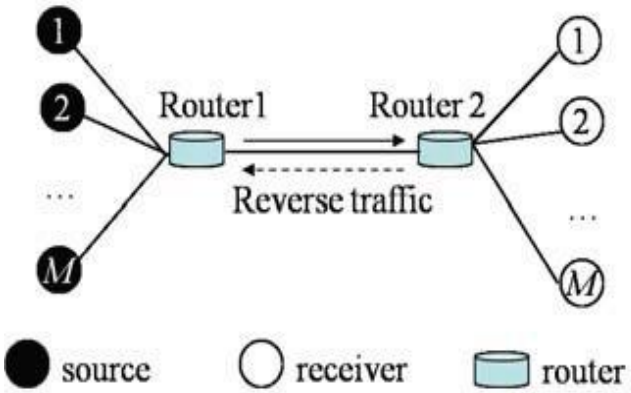


Fig. 3. Simulation network.

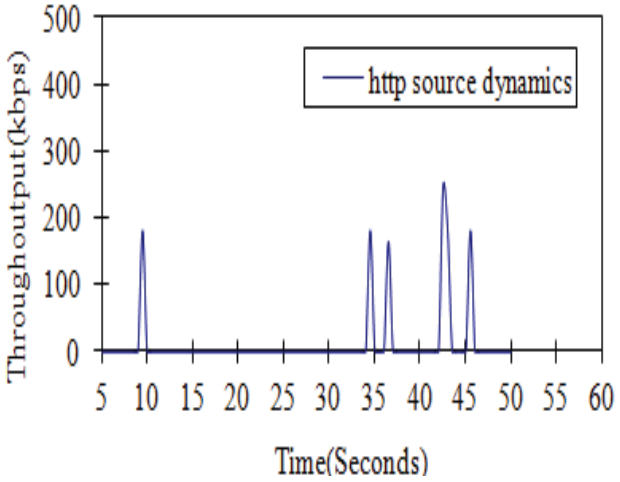
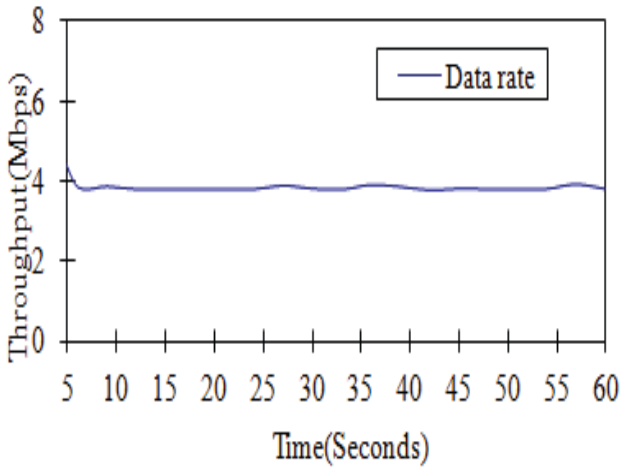


Fig. 4. Http sessions example.

The TBO and the buffer capacity B in Router 1 in each experiment are set according to the approaches discussed in Section III. We also adopt some typical values from the experiments of existing works so that we can make our experiments more meaningful. In particular, all the ftp packets have the same size of 1024 bytes [19] while the http packet size is uniformly distributed in the range of [800, 1300] bytes [12]. In order to demonstrate and discuss the robustness of our IntelRate controller, our experiments would focus on the testing of the 100 long-lived ftp flows, unless otherwise stated. The 100 sporadic short-lived http flows just act as the disturbance to the ftp traffic and their transfer size follows the real web traffic scenario; it has a Pareto distribution [50] with a mean transfer size of 30 packets [6]. The arrivals of http flows follow a think-time [50] uniformly distributed in [0.1s , 30s]. One of the http session examples is shown in Fig. 4. The uncontrolled ftp flow keeps using a window size of 100 packets and operates at an almost constant rate as shown in Fig. 5. These http and UDP-like flows generate an aggregate traffic v(t) as discussed in Section II. The experiments were conducted using OPNET Modeler 11.5 [51] on an Intel Core TM2 Quad platform with 2.40GHz processor. Typical simulated time is set according to the specific experiment scenario.



\

Fig. 5. Uncontrolled ftp flow.

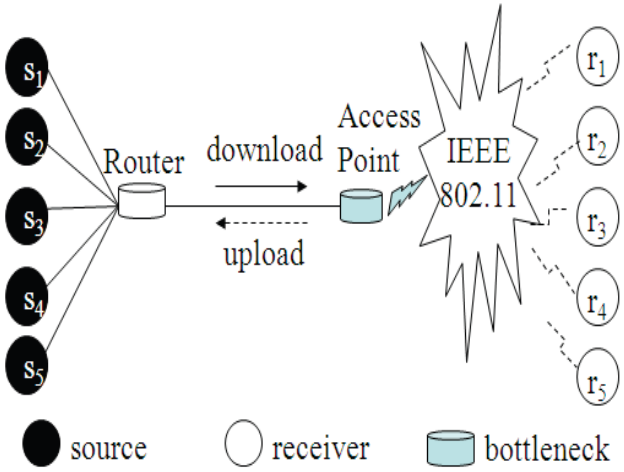


Fig. 6. Wireless LAN.

The simulation time depends on bottleneck bandwidth and the simulated time. A typical simulation run usually takes hours or days. For example, in order to observe the source throughput behavior before and after the network parameter change, we set a longer simulated time for such an experiment than a max-min fairness experiment. The number of packets generated in an experiment is related to the TBO value, the bandwidth, the simulated time and the traffic intensity. The controller is evaluated by the following performance measures.

- Source throughput (or source sending rate) is defined to be the average number of bits successfully sent out

by a source per second, i.e. bits/second [51]. Here, a bit is considered to be successfully sent out if it is part of a packet that has been successfully sent [51].

- IQSize is the length of the bottleneck buffer queue (measured in packets) seen by a departing packet [52].
- Queueing delay is the waiting time of a packet in therouter queue before its service. Measurements are taken from the time the first bit of a packet is received at the queue until the time the first bit of the packet is transmitted.
- Queueing jitter is the variation of queueing delay due to the queue length dynamics, and is defined as the variance of the queuing delay.
- Link (or bottleneck) utilization is the ratio between the current actual throughput in the bottleneck and the maximum data rate of the bottleneck. It is expressed as a fraction less than one or as a percentage.
- Packet loss rate is the ratio between the number of packet dropped and the number of total packets received per second by the bottleneck.
- Max-min fairness: A feasible allocation of rates is 'maxmin fair' if and only if an increase of any rate within the domain of feasible allocations must be at the cost of a decrease of some already smaller or equal rates [1].

III. COMPARISON

Our preliminary results in [53], [54] demonstrate that the IntelRate controller is superior to other explicit congestion controllers. For examples, the IntelRate controller has better robustness and link utilization than the XCP upon bandwidth variations; it has a lower requirement on computational and memory resources than API-RCP while having equivalent and even better performances (the comparisons of the computational intensity and memory requirement with other controllers will be presented in our other papers). Here we pick the QFCP and Blind (we didn't choose ErrorS or MAC from the same paper to do the comparison because ErrorS is an interchangeable algorithm of Blind and faces the same problem while MAC is too complicated to be put into practice so far) for further comparison as they are enhancements of XCP in reducing the bandwidth estimation errors as discussed before.

IV. CONCLUSION

A fuzzy logic control based QoS management approach has been proposed for WSANs. With this approach, the sampling period of each source sensor node is adjusted dynamically so that the deadline miss ratio associated with the relevant data transmission from the sensor to the actuator is maintained at a desired level. In this way, QoS requirements with respect to timeliness, reliability, and robustness can be satisfied. Simulation results have demonstrated the effectiveness of the proposed approach. To verify the effectiveness and superiority of the IntelRate controller, extensive experiments have been conducted in OPNET modeler. In addition to the feature of the FLC being able to intelligently tackle the nonlinearity of the traffic control systems, the success of the IntelRate controller is also attributed to the careful design of the fuzzy logic elements.

Future Work: Our future work in this direction includes: 1) improvement of the FLC-QM scheme for large-scale WSANs through, e.g., developing a unified framework; 2) extensive simulation studies on WSANs with more complex network topology; and 3) experimental studies and practical implementation of the FLC-QM scheme in WSANs.

V. REFERENCES

[1] M. Welzl, Network Congestion Control: Managing Internet Traffic. John Wiley & Sons Ltd., 2005.

[2] R. Jain, "Congestion control and traffic management in ATM networks: recent advances and a survey," Computer Networks ISDN Syst., vol. 28, no. 13, pp. 1723–1738, Oct. 1996.

[3] V. Jacobson, "Congestion avoidance and control," in Proc. 1988 SIGCOMM, pp. 314–329.

[4] V. Jacobson, "Modified TCP congestion avoidance algorithm," Apr. 1990.

[5] K. K. Ramakrishnan and S. Floyd, "Proposals to add explicit congestion notification (ECN) to IP," RFC 2481, Jan. 1999.

[6] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," in Proc. 2002 SIGCOMM, pp. 89–102.

[7] S. H. Low, F. Paganini, J.Wang, et al., "Dynamics of TCP/AQM and a scalable control," in Proc. 2002 IEEE INFOCOM, vol. 1, pp. 239–248.

[8] S. Floyd, "High-speed TCP for large congestion windows," RFC 3649, Dec. 2003.

[9] W. Feng and S. Vanichpun, "Enabling compatibility between TCP Reno and TCP Vegas," in Proc. 2003 Symp. Applications Internet, pp. 301–308.

[10] M. M. Hassani and R. Berangi, "An analytical model for evaluating utilization of TCP Reno," in Proc. 2007 Int. Conf. Computer Syst. Technologies, p. 14-1-7.

[11] N. Dukkipati, N. McKeown, and A. G. Fraser, "RCP-AC congestion control to make flows complete quickly in any environment," in Proc. 2006 IEEE INFOCOM, pp. 1–5.

[12] Y. Zhang, D. Leonard, and D. Loguinov, "JetMax: scalable max-min congestion control for high-speed heterogeneous networks," in Proc. 2006 IEEE INFOCOM, pp. 1–13.

[13] B. Wydrowski, L. Andrew, and M. Zukerman, "MaxNet: a congestion control architecture for scalable networks," IEEE Commun. Lett., vol. 7, no. 10, pp. 511–513, Oct. 2003.

[14] Y. Zhang and M. Ahmed, "A control theoretic analysis of XCP," in Proc. 2005 IEEE INFOCOM, vol. 4, pp. 2831–2835.

[15] Y. Zhang and T. R. Henderson, "An implementation and experimental study of the explicit control protocol (XCP)," in Proc. 2005 IEEE INFOCOM, vol. 2, pp. 1037–1048.

Author's Profile:

Kishore Teppala, PG Scholar, (Department of Software Engineering, Kakinada Institute of Engineering &Technology (KIET), JNTUK A.P.,Email: kishore.yfc@gmail.com)

K.Ravi Kumar, He has completed his master degree in CSE at Pragathi Engineering College, surampalem. He has 8 years of teaching experience in KIET. Currently working as associate professor and HOD in KIET. He has guided more than 10 projects for post graduates and more than 15 projects for graduates. His research areas are bioinformatics, datamining and wireless sensor networks.