An efficient AFRA using an Intelligent Fuzzy Repertory Table technique

Vinothini S[1], Chandra Segar Thirumalai[2], Vijayaragavan R[3]

[1]*Student, CK College of Engineering & Technology, Anna University, Cuddalore, India.*
[2] *Assistant Professor Senior, School of I.T and Engg., VIT University, Vellore, India.*
[3]*Associate Professor, School of Advanced Sciences, VIT University, Vellore, India.*

---***---

Abstract - *Today routing algorithm offers a huge advantage such as shaping traffic, congestion control, flow control, bandwidth, processing delay, hop count, MTU (Maximum Transmission Unit), reliability, etc., In the real world, many systems developed by considering the above factors to route over the best path among the source and destination. This leads to the need of evaluation over various routing algorithms such as distance vector, link state and OSPF. This paper addresses with an expert system called AFRA (Advanced Fuzzy Routing Algorithm) which is to itinerary the packet in an efficient way and to amplify the degree of network utilization. Moreover, this model primarily designed for; (i) To reduce the route exploration and (ii) To reduce the extra overhead over the heavily loaded systems by quantifying various routing metrics.*

Key Words: Dynamic Load Balancing, Trapezoidal Fuzzy Repertory Table, Trapezoid Number

1. INTRODUCTION

In the recent days the usage of hosts over the network has been increased. Many network researchers and practitioners are still undergoing in the area of load balancing management metrics to reduce traffic, collision, packet delivery failures and to develop practices that consistently generate better outcomes. Load balancing [1] plays a vital role to minimize latency for a packet between client and server over the heavily loaded network systems. Dynamic load balancing policies [2], [3], [4], [5], [6] present the possibility of improving load distribution at the cost of improving performance, flexibility, reliability, scalability and availability. The operating cost of dynamic load balancing may be large [7], to a huge heterogeneous distributed system. Among the Static load balancing policies and Dynamic load balancing Zhang et al. [8] shown that the static load balancing policies are more desirable when the system loads are light and fair or when the overhead is not insignificantly high. This paper went into the dynamic load balancing which may also facilitate us to distribute among various network systems and make a parametric tuning to develop the system performance, flexibility, reliability, scalability and availability.

Here we have projected a new technique called FRT technique to minimize the traffic and latency by means of quantifying various routing metrics [4] like packet delivery ratio (pdr), routing overhead, end to end delay. Latency is a measure of time delay over the communication systems. In addition to that this technique can also be applied on both hop by hop and end to end traffic managements. Some of the well known hop by hop applications are RSVP, IGMP, 802.1z, 802.1p/Q, etc., which uses stateless load balancer and some of the well known end to end applications are ATM, TCP, HTTP, FTP, SMTP, etc., which uses stateful load balancer.

2. TRAPEZOIDAL FRT

Dr.Lotfi A. Zadeh, is the father of fuzzy sets and fuzzy logic, in 1965. Fuzzy sets are generalized sets such that the membership is a real number in the [0, 1] range instead of 0 and 1 only.

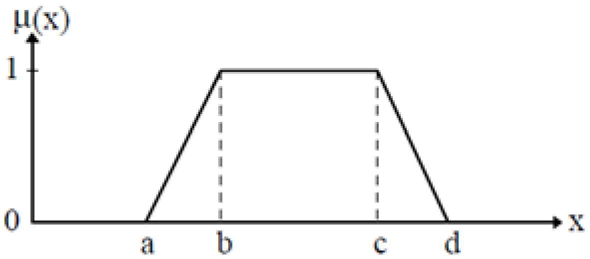

Fig. 2: Fuzzy Trapezoidal Graph

Fuzzy logic proved to be a very powerful concept in the various disciplines, and industries applications. Dynamic load balancing policies is probably aimed to improving load distribution at the cost of high performance, flexibility, reliability, scalability and availability.

2.1. Rating attributes using trapezoid numbers

A fuzzy repertory table (FRT) also looks like a rectangular matrix with elements (as columns) and constructs (as rows). Each row–column intersection contains a rating. Such a rating is a trapezoid number showing how a user applied a given construct to a particular element. A trapezoid number (a,b,c,d) is a fuzzy set that has a membership function of the following form:

$$\mu(x)=\begin{cases} 0 & if \quad x \le a, \\ \frac{x-a}{b-a} & if \quad a < x < b, \\ 1 & if \quad b \le x \le c, \\ \frac{d-x}{d-c} & if \quad c < x < d, \\ 0 & if \quad x \ge d. \end{cases}$$

By using trapezoid numbers, the FRT technique [9], [10], [11], [12], lighten up the restriction, of the classical repertory grid technique, that the ratings must be crisp numbers in a predefined range. Moreover, trapezoid numbers enable the FRTs to provide categorical and numerical data types that may be given by means of linguistic terms. For instance, in the FRT developed in our Load Balancing scenario, taking traffic attribute F1 as an example, packet delivery ratio of a routing port is rated on a 1–4 rating scale (this rating provides an indication of packet forwarding preferences: 1— Less traffic, 2 — Moderate traffic 3— High traffic and 4—Very High traffic).

There are totally five different scales are used in software to quantify a metric such as Nominal, Ordinal, Interval, Ratio and Absolute. In the FRT table, each value is expressed by a membership function that is determined from the construct type and direct interaction with the user.

Unordered-discrete or nominal scale:
There are no unique numbers or strings. Just to define the elements labeling or naming is used.
E.g: router make name, ethernet cable name, etc.,
Ordered-discrete or ordinal scale:
A number states precisely an element's position in the series established by the scale
E.g: packet number, acknowledgement number, etc.,
Crisp interval value:
The user assigns two numbers, x and y, to an element in such a way that the interval between these two values is meaningful for him. The trapezoidal function associated with this value is a=b= x & c=d= y.
Boolean:
Checks the attributes whether it exists or not.
E.g: connection checks, success or failure;
Absolute scale:
The user assigns a number, x, to an element. The function associated with this value is one with the parameters a = b = c = d = x.
E.g:, age, date, weight, etc.,

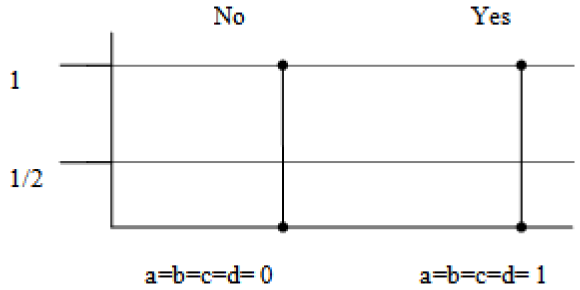

Fig. 2.1: Boolean Representation

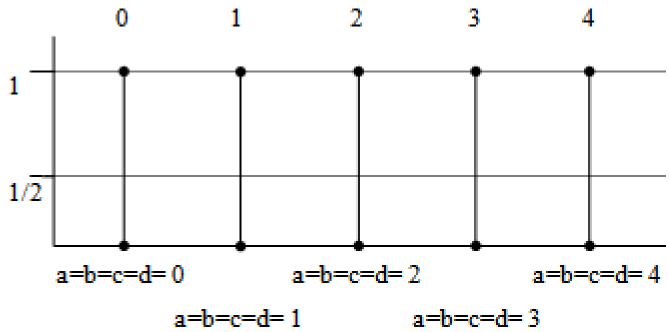

Fig. 2.2: Fixed Value Rating from 0 to 4

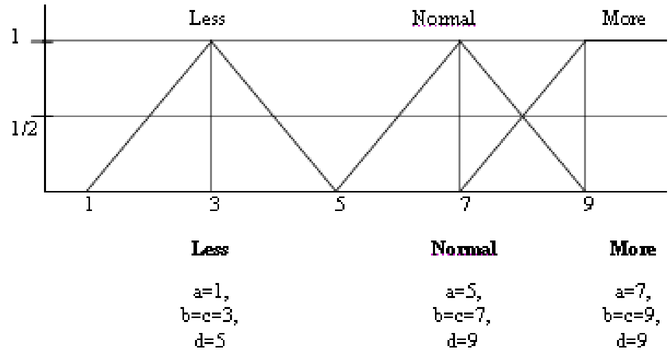

Fig. 2.3: Rating for the attribute

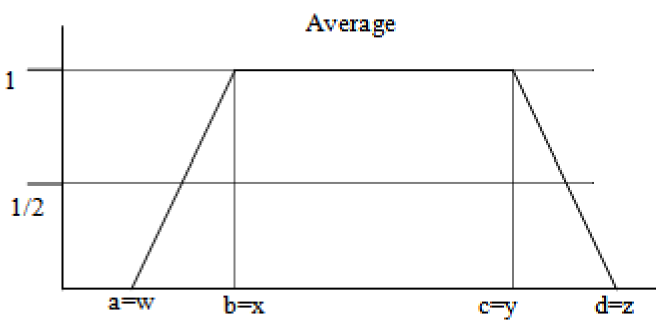

Fig. 2.4: Continuous fuzzy value for Trapezoidal Representation

Table 2.1: Rating attributes using trapezoid numbers

Set	Complexity Attribute	Rating Scheme	Assignments for the Trapezoid Numbers
F_1	Hop count	Ranking using crisp values range from 0 to 255	a=b=c=d=0, a=b=c=d=8 a=b=c=d=9, a=b=c=d=15 a=b=c=d=16, a=b=c=d=255
F_2	Size of the load balancing system (in LOC)	Ordered discrete type – Any numerical value of count	Low: a=500, b=1000, c=1500 and d=2000 Average: a=1500, b=2000, c=2500, and d=3000 High: a=2500, b=3000, c=3500, and d=4000 Very High: a=4000, b=4500, c=5000, and d=5000
F_3	Bandwidth Utilization	Ranking using crisp values range from 0 to 5	a=b=c=d=0, a=b=c=d=1, a=b=c=d=2, a=b=c=d=3, a=b=c=d=4, a=b=c=d=5
F_4	End to end delay	Ranking using crisp values range from 0 to 5	a=b=c=d=0, a=b=c=d=1, a=b=c=d=2, a=b=c=d=3 a=b=c=d=4, a=b=c=d=5
F_5	Routing Overhead	Ranking using crisp values range from 0 to 255	a=b=c=d=0, a=b=c=d=8, a=b=c=d=9, a=b=c=d=15, a=b=c=d=16, a=b=c=d=255
F_6	Processing Delay	Ordered discrete type – Any numerical value of count	Less : a=1,b=c=3, and d=5 Normal : a=5,b=c=7, and d=9 More : a=7,b=c=9, and d=9
F_7	Queuing Delay	Ordered discrete type – Any numerical value of count	Less : a=1,b=c=3, and d=5 Normal : a=5,b=c=7, and d=9 More : a=7,b=c=9, and d=9
F_8	Throughput of the system with respect to the load balancing algorithm	Ranking using crisp values range from 0 to 5	a=b=c=d=0, a=b=c=d=1, a=b=c=d=2, a=b=c=d=3, a=b=c=d=4, a=b=c=d=5

3. RELATED WORK

Many network researchers and practitioners are still undergoing in the area of load balancing management metrics to reduce traffic, collision, packet delivery failures and to develop practices that consistently generate better outcomes. In this section, study has been made on various load balancing algorithm for the assessment of routing metrics that are used in the development of traffic management. Most metric sets deal with a variation of packet delivery ratio (pdr), routing overhead, end to end size, etc.,

Several distributed load balancing models were proposed in the literature: the gradient model [4], [13], [11], sender or receiver-initiated diffusion [9], [16], [17], the hierarchical balancing model [17], and others. Practical implementations of such models vary based on the resources considered for scheduling: CPU [18], [8], [20], [3], [22], memory [1], [24], [25], [26], or combinations of I/O, CPU and memory [4], [5], [29]. Of particular interest for us are load balancing systems used for distributed Internet services.

Gao [2] proposed a category of load balancing algorithms which works on the hardware system contains identical CPU. Each and every processor shares its load with one another to shred its load. In order to migrate the load among the identical processor the job dispatcher plays a vital role by measuring the status of the processor workload and assigns the new task to the processor which has less or idle workload. This assignment of workload among the processors requires high numerical calculations and relays on the huge support of networks.

Ni [16] proposed a drafting algorithm for dynamic task migration. It is observed that the processors do not need to communicate precise numerical load measurements for a dynamic load balancing scheme to be effective. Therefore, a 3-level (heavy, normal, and light) system is used. A processor communicates only with a group of processors called the candidate processors. A lightly loaded processor requests a heavily loaded candidate processor to send a bid for task migration. A task is migrated from the heavily loaded processor after the lightly loaded processor has sent a Select message to it. Since broadcasting the load at every change in load level may create too much communication traffic (and hence a longer response time), piggybacking is recommended to reduce the number of messages. However, unless a processor has every other processor in the system as its candidate processor, a lightly loaded processor may not notice the existence of some heavily loaded processor. Therefore, this scheme guarantees that every possible task migration will be carried out only when all load messages are broadcasted to every other processor in the system.

4. TRAPEZOIDAL FRT IN LOAD BALANCING

In this work, our approach is constructed with the inspiration of a communication model observed in FRT (fuzzy repertory table) combined with the capabilities of the fuzzy logic technique. The projected algorithm first determines the crisp path rankings for all eligible paths between the source and destination nodes from the viewpoint of fuzzy inference. The path with the highest ranking is then chosen to route the traffic flow. The path congestion rate in this paper represents the degree of the path usability in the sense of the multiple criteria required. Whenever traffic flow is routed to a chosen path, a packet is dropped when it arrives at a full buffer. The fuzzy Inputs are chosen as the traffic rate, bandwidth, throughput, end to end delay...etc based on the metric we used in the table. The fuzzy output is load balancing, shaping traffic.

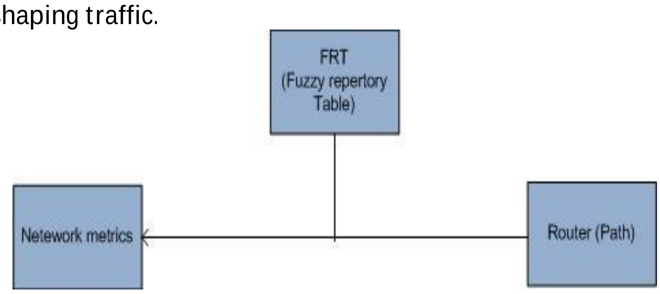

Fig. 4.1: FRT interface b\w Network metrics and Router

In recent days, the importance communication networks all are well known; like increasing online social networks such as Friendster, MySpace, or the Facebook have experienced *exponential* growth in membership in recent years. These networks over attractive means for interaction and communication, these type sites creating more traffic in networks, these are the heart of traffic in networks, in this situation we have to identifying the best path to transfer the packets in efficient way is a mandatory process for the client and servers.

Identification of path with enough quality is a tedious process in any network because of vast number of factors that affects it. Also identifying path delay, path utilization, shaping traffic, congestion control, flow control, bandwidth, processing delay, hop count, MTU (maximum transmission unit), reliability etc. There are numerous amounts of metrics (attributes) available to rank the Path to send the packet on that path, but the difference is usage of technique and Factors/attributes. Elements and constructs (dimensions of similarity and differences between elements) are central to knowledge representation in repertory grids. The most basic form of a repertory grid is a rectangular matrix with elements as columns and constructs as rows.

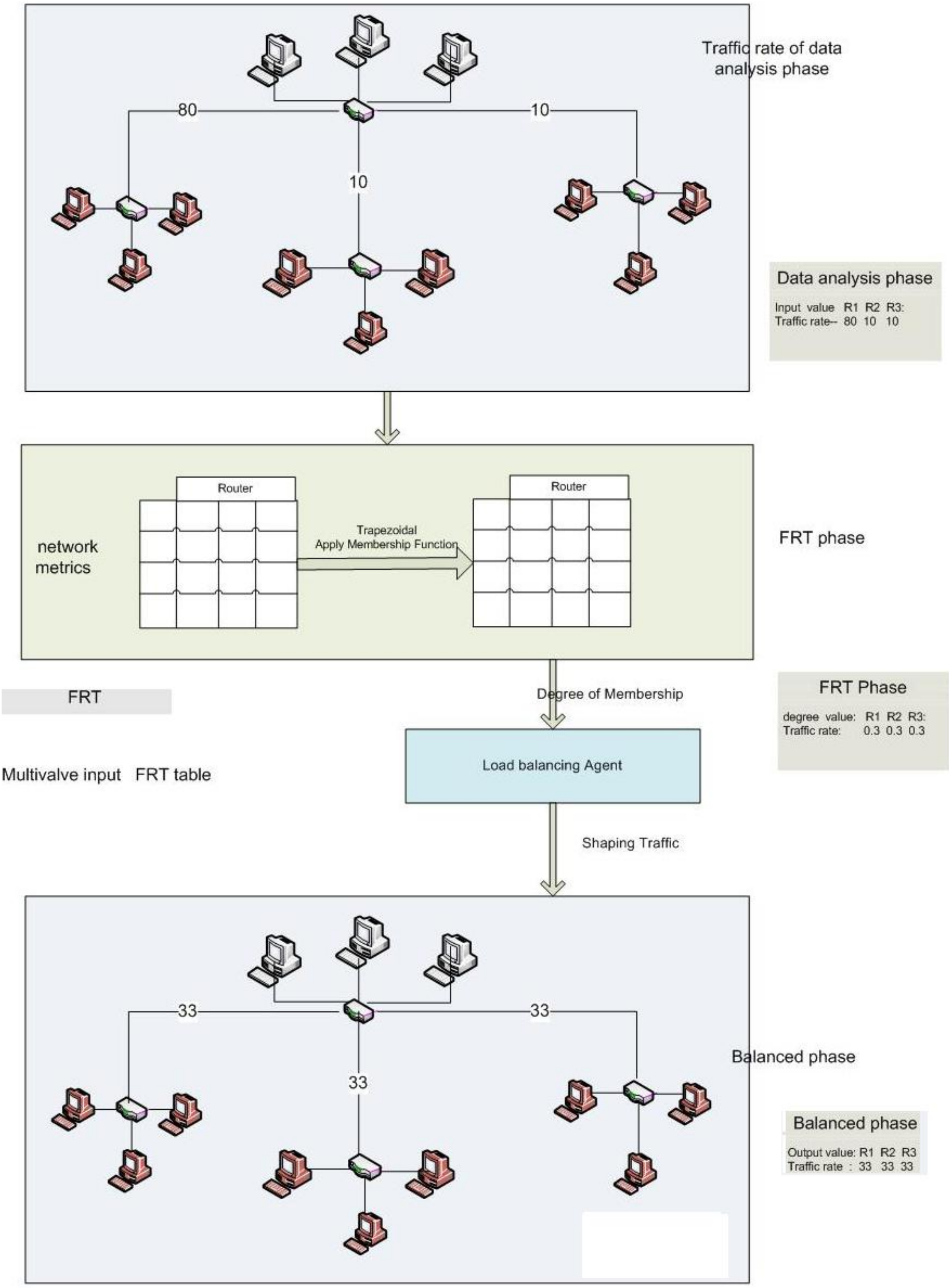

Fig. 4.2: Basic View of Projected Fuzzy Load Balancing Architecture

Table 4.1: FRT process on Construct and Element

	Construct1	Construct2	.	Construct n
Element1				
Element2				
. .				
Element n				

Each row-column intersection in the grid contains a rating to show how a person applied a given construct to a particular element. Thus, if the element is closest to the left pole of the construct, he places a tick; otherwise, a cross. Within classification problems, the elements of a repertory grid will be those classes which we want to learn to classify and the constructs will be the input variables, whose different values can distinguish a class from another one.
Since we are interested in to obtain the input variables plus their definition domains (the rating used by the expert to value each input variable) that the expert uses in a classification task and in the same way in which he uses them, for facilitating their integration into the knowledge acquisition process, we will need to make several changes to the classic repertory grid.
To realize the result we have taken the free source of network simulator 2 (ns-2) as a standard simulation package and extended it to implement our advanced fuzzy routing algorithm with OSPF, distance vector, link state routing algorithm. The aim of the simulator is to closely mirror the essential features of the concurrent and distributed behavior of a generic communication network without sacrificing efficiency and flexibility in code development.

NS2 configuration of AFRA:

```
set val(chan)  Channel/WirelessChannel
set val(prop)  Propagation/TwoRayGround
set val(netif) Phy/WirelessPhy
set val(mac)   Mac/802_11
set val(ifq)   Queue/DropTail/PriQueue
set val(ll)    LL
set val(ant)   Antenna/OmniAntenna
set val(ifqlen) 50
set val(nn)    5
set val(rp)    AFRA
```

```
//AFRA trapezoid variables assignment:
for (i=0;i<100;i++)
  if (bandwidth[i]!=0)
   { if(bandwidth[i]>=0 && bandwidth[i]<=1500)
    {     degree[i]=0;
          printf("\nDegree of Node %d is:  %f -- x=%d [1]", i,degree[i],bandwidth[i]);  }
   else if(bandwidth[i]>=1500 && bandwidth[i]<=2000)
   {      degree[i]= ((float)(bandwidth[i] - 2000)/(float)(1500-2000));
          printf("\nDegree of Node %d is:  %f-- x=%d [2]",i,degree[i],bandwidth[i]);   }
   else if(bandwidth[i]>=2000 && bandwidth[i]<=3500)
   {      degree[i]= 1;
          printf("\nDegree of Node %d is:  %f-- x=%d [3]", i,degree[i],bandwidth[i]);   }
   else if(bandwidth[i]>=3500 && bandwidth[i]<=6000)
   {      degree[i]=((float)6000-bandwidth[i])/((float)6000-3500);
          printf("\nDegree of Node %d is:  %f-- x=%d [4]",i,degree[i],bandwidth[i]);   }
  else  {  degree[i]= 0;
           printf("\nDegree of Node %d is:  %f-- x=%d [5]",i,degree[i],bandwidth[i]);   }
        }
```

5. CONCLUSIONS AND FUTURE WORK

Our fuzzy repertory table based routing algorithm metrics result will be compared with other routing algorithms such as Distance Vector, Link State & OSPF to run on the network topology. The expected results will indicate that the projected algorithm does a better job at dispersing traffic in a more uniform manner throughout the network. In addition to that it will also handles an increased traffic load as well as decreased transmission delay by utilizing network resources more efficiently. The advantages of such an intelligent algorithm include increased flexibility in the constraints that can be considered together in making the routing decision efficiently and likewise the simplicity in taking into account multiple constraints.

In the near future the next generation networks will have capabilities including soft-switches, which allow such an intelligent technique -based routing algorithm to shapes the traffic & load balancing autonomously, and then they can be substituted with the conventional routing algorithms. As part of future research, we will study in detail different load metrics that can be used to evaluate the actual load of a routing node so we can obtain a fairer share of load among neighboring routing nodes. We will also study how to apply new FRT dynamic load balancing algorithm mechanisms to perform different resource management strategies, such as the frequency assignment problem, or controlling the number of hops transmit power control in both hop by hop and end to end traffic managements.

REFERENCES

[1] A. Acharya and S. Setia, ªAvailability and Utility of Idle Memory in Workstation Clusters,º Proc. ACM SIGMETRICS Conf. Measuring and Modeling of Computer Systems, May 1999.

[2] C.Gao, J.W.S.Liu and M.Railey, "Load Balancing Algorithms in Homogeneous Distributed Systems", Proceedings of the 1984 International Conference on Parallel Processing, CRC Press, Bocaraton, Fl, August 1984, pp.302-306.

[3] C. Hui and S. Chanson, ªImproved Strategies for Dynamic Load Sharing,º IEEE Concurrency, vol. 7, no. 3, 1999.

[4] D. Andresen and T. Yang, ªSWEB++: Partitioning and Scheduling for Adaptive Client-Server Computing on WWW,º Proc. 1998 SIGMETRICS Workshop Internet Server Performance, June 1998.

[5] D. Andresen, T. Yang, O. Ibarra, and O. Egecioglu, ªAdaptive Partitioning and Scheduling for Enhancing WWW Application Performance,º J. Parallel and Distributed Computing, vol. 49, no. 1, Feb. 1998.

[6] D.L. Eager, E.D. Lazowska, and J. Zahorjan, "A Comparison of Receiver-Initiated and Sender-Initiated Adaptive Load Balancing," Performance Evaluation, vol. 6, pp. 53-68, 1986.

[7] D.L. Eager, E.D. Lazowska, and J. Zahorjan, "Adaptive Load Sharing in Homogeneous Distributed Systems," IEEE Trans. SoftwareEng., vol. 12, no. 5, pp. 662-675, May 1986.

[8] D.P. Bertsekas and J.N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods. Prentice-Hall, 1989.

[9] F. Douglis and J. Ousterhout, ªTransparent Process Migration: Design Alternatives and the Sprite Implementation,º Software-Practice and Experience, vol. 46, no. 2, 1997.

[10] F. Muniz and E.J. Zaluska, ªParallel Load Balancing: An Extension to the Gradient Model,º Parallel Computing, vol. 21, pp. 287-301, 1995.

[11] F.C.H. Lin and R.M. Keller, "The Gradient Model Load Balancing Method," IEEE Trans. Software Eng., vol. 13, no. 1, pp. 32-38, Jan. 1987.

[12] G. Voelker, ªManaging Server Load in Global Memory Systems, Proc. ACM SIGMETRICS Conf. Measuring and Modeling of Computer Systems, May 1997.

[13] Jose J, Juan Luis Castro and Jose Manuel Zurita., Fuzzy Repertory Table: A method for acquiring knowledge about input variables to machine learning algorithm, IEEE Transactions on Fuzzy System, Vol 12, No.1, February 2004.

[14] Jose J, Nicholas R, Xudong Luo., Acquiring domain knowledge for negotiating agents: a case of study, Elsevier, 22 september, 2003.

[15] K.G. Shin and Y. Chang, "A Coordinated Location Policy for Load Sharing in Hypercube-Connected Multicomputers," IEEE Trans.Computers, vol. 44, no. 5, pp. 669-682, May 1995.

[16] L.M.Ni, C.W.Xu and T.B.Gendreau, "A Distributed Drafting Algorithm for Load Balancing," IEEE Trans. on Software Engg., Vol.SE-13, No.10, October1985, pp.1153-1161.

[17] L. Xiao, X. Zhang, and Y. Qu, ªEffective Load Sharing on Heterogenous Networks of Workstations,º Proc. 14th Int'l Parallel and Distributed Processing Symp. (IPDPS 2000), May 2000.

[18] M. Harchol-Balter and A. Downey, ªExploiting Process Lifetime Distributions for Load Balancing,º ACM Trans. Computer Systems, vol. 3, no. 3, 1997.

[19] M. Harchol-Balter and A.B. Downey, "Exploiting Process Lifetime Distributions for Dynamic Load Balancing" Proc. 1996 ACM SIGMETRICS, pp. 13-24, Philadelphia, May, 1996.

[20] M. Willebeck-LeMair and A. Reeves, ªStrategies for Dynamic Load Balancing on Highly Parallel Computers,º IEEE Trans. Parallel and Distributed Systems, vol. 4, no. 9, pp. 979-993, Sept. 1993

[21] N. Carriero, E. Freeman, D. Gelernter, and D. Kaminsky, Adaptive Parallelism and Piranha,º Computer, vol. 28, no. 1, pp. 40-49, Jan. 1995.

[22] Narayanan Prasanth N, S. Ganesh , G. Arul Dalton, Prediction of Maintainability using Software Complexity Analysis: an Extended FRT, ICCCN 2008 International Conference on Computing, Communication and Networking, Vol 1, Issue 1, 18-20 Dec 2008 pages 1-9.

[23] R. Luling, B. Monien, and F. Ramme, Load Balancing in Large Networks: A Comparative Study, Proc. Third IEEE Symp. Parallel and Distributed Processing, pp. 686-689, Dec. 1991.

[24] Rathod, P., 1981. Methods for the analysis of repertory grid data. Personal Construct Psychology. Recent Advances in the theory and practice. St. Martins Press, Newyork.

[25] S. Chen, L. Xiao, and X. Zhang, ªDynamic Load Sharing with Unknown Memory Demands of Jobs in Clusters,º Proc. 21st Ann. Int'l Conf. Distributed Computing Systems (ICDCS 2001), pp. 109-118, 2001.

[26] T. Kunz, ªThe Influence of Different Workload Descriptions on a Heuristic Load Balancing Scheme,º IEEE Trans. Software Eng., vol. 17, no. 7, July 1991.

[27] Chandramowliswaran N, Srinivasan.S and Chandra Segar.T, "A Note on Linear based Set Associative Cache address System International J. on Computer Science and Engg. (IJCSE) & India Engineering Journals & 0975-3397 Vol. 4 No. 08 / pp. 1383-1386 / Aug. 2012

[28] V.A. Saletore, ªA Distributed and Adaptive Dynamic Load Balancing Scheme for Parallel Processing of Medium-Grain Tasks,º Proc. Fifth Distributed Memory Computing Conf., pp. 995-990, Apr. 1990.

[29] Chandra Segar T, Vijayaragavan R, "Pell's RSA key generation and its security analysis IEEE Computing, Communications and Networking Technologies (ICCCNT), India IEEE & 978-1-4799-3925-1 Page 1 – 5/July 2013

[30] X. Zhang, Y. Qu, and L. Xiao, ªImproving Distributed Workload Performance by Sharing both CPU and Memory Resources, Proc. 20th Int'l Conf. Distributed Computing Systems (ICDCS 2000), Apr. 2000.

[31] Chandramowliswaran N, Srinivasan.S and Chandra Segar.T, "A Novel scheme for Secured Associative Mapping" The International J. of Computer Science and Applications (TIJCSA) & India TIJCSA Publishers & 2278-1080 Vol. 1, No 5 / pp. 1-7 / July 2012

[32] Y. Zhang, H. Kameda, and K. Shimizu, "Adaptive Bidding Load Balancing Algorithms in Heterogeneous Distributed Systems," Proc. IEEE Second Int'l Workshop Modeling, Analysis, and Simulation of Computer and Telecomm. Systems, pp. 250-254, Durham, N.C., Jan. 1994.

[33] Y. Zhang, K. Hakozaki, H. Kameda, and K. Shimizu, "A Performance Comparison of Adaptive and Static Load Balancing in Heterogeneous Distributed Systems," Proc. IEEE 28th Ann. Simulation Symp., pp. 332-340, Phoenix, Ariz., Apr. 1995.

[34] Zhou, and Songnian, "Performance Studies of Dynamic Load Balancing in Distributed Systems," US Berkely EECS, CSD-87-376, October 1987.

BIOGRAPHIES

Vinothini S, completed M.E. in 2012 under the major of Applied Electronics with distinction from C.K college of Engineering and Technology and B.E in Electronics and Communication Engineering affiliated to Anna University. Her area of specialization includes Fuzzy Systems, Digital Image Processing, Computer Networks.

Prof. Chandra Segar T, currently working as Assistant Prof. Senior in School of Information and Technology and Engineering, VIT University, Vellore, India. His area of specialization includes Linear Cryptanalysis, Public Key Cryptosystems, Fuzzy Systems, Automata and Networking. About his publication, currently holds six International journals and one International conference.

Prof. Vijayaragavan, Ph.D is currently working as Associate Professor in VIT University, Vellore, India as Associate Professor in School of Advanced Sciences. His area of interest includes Cryptosystems, etc., About his publication, more than fifteen International journals and several International conferences.

