LOAD BALANCING IN WIRELESS AD-HOC NETWORKS WITH LOW FORWARDING INDEX

Reena Dadhich[1] & Aditya Shastri[2]

[1]Department of MCA, Engineering College Ajmer, India
reena.dadhich@gmail.com
[2]Department of Computer Science, Banasthali University, India.
adityashastri@yahoo.com

Abstract

A wireless ad-hoc network comprises of a set of wireless nodes and requires no fixed infrastructure. For efficient communication between nodes, ad-hoc networks are typically grouped in to clusters, where each cluster has a clusterhead (or Master). In our study, we will take a communication model that is derived from that of BlueTooth. Clusterhead nodes are responsible for the formation of clusters each consisting of a number of nodes (analog to cells in a cellular network) and maintenance of the topology of the network. Consequently, the clusterhead tend to become potential points of failures and naturally, there will be load imbalanced. Thus, it is important to consider load balancing in any clustering algorithm. In this paper, we consider the situation when each node has some load, given by the parameter forwarding Index.

Keywords

Ad-hoc Networks, Clustering, Bluetooth, Forwarding Index,Algorithm.

1. INTRODUCTION

Ad-hoc networks are expected to play a significant role in the future mobile computing applications. A wireless ad-hoc network consists of a set of self-organizing mobile nodes which required no fixed infrastructure and which communicate with each other over wireless links. For efficient communication between nodes, ad-hoc networks are typically grouped into clusters, where each cluster has a clusterhead (or Master). Communication between nodes in different clusters is through gateway nodes; these are also known as bridge nodes. Bluetooth is an emerging technology for indoor wireless picocellular environment and it employs a master-slave model for communication between nodes. In this model each cluster has a star topology, with a master at the center of the star, and the Master controls the traffic to the Slaves. In order to streamline flow of information between nodes and to adapt to topological changes, the entire network is divided into cluster of nodes.

Efficient clustering and topology construction algorithms play a very important role in the fast connection establishment of ad-hoc networks. The performance of these networks is chiefly dependent on the device discovery time, i.e. the time taken by a node to discover and to connect to another node in its radio range which is already part of the existing network. This device discovery time is also crucial in other situations. For example, when a large number of devices within radio range of each other are powered on, the time taken to complete the formation of the network is an important performance criterion.

DOI : 10.5121/ijwmn.2011.3104

Throughout the paper we work with Bluetooth model which is a synchronous system in which every node has a unique Id, but does not know the ID of any other node. In this paper we study the Randomized algorithm for cluster formation, due to Aggarwal et. al. [9] for asynchronous complete networks of N nodes (all within radio range of each other), which can be used to construct a minimal set of star-shaped clusters of limited size having the forwarding index within at most a constant factor of the optimal.

Most of the existing algorithms focus on partitioning the network into clusters and differ mainly in the clusterhead election criterion. Further, they also do not consider the stability of the network while clustering. As a result, these nodes take greater responsibility and thus their energy gets depleted faster making them to drop out. This situation creates congestion in the network because large number of routes passes through the clusterhead node.

Thus, the clustering algorithm should guarantee that the extra workload is always balanced between all the nodes of the network. It other words, the responsibility of acting as clusterheads should be fairly distributed in the network. It will also be bad from the fault-tolerance point of view for if such a node were to fail a large part of the network would come to a halt. Therefore, there is a need to evenly distribute the routing [9] & load among all nodes in the cluster (i.e., load balancing) [1].

In this paper, we propose a parameter; called forwarding Index for ad-hoc networks. Forwarding Index is a measure of routing to be evenly distributed proposed in [2]. The vertex (edge) forwarding index of the network is the minimum value of the largest load occurring at a vertex (edge) taken over all routings, where load of a vertex (edge) is defined as the number of routes passing through that vertex or edge. The problem of determining the forwarding Index of a network is NP-complete [2,3,6].

In ad-hoc networks the value of forwarding Index for clusterhead node is maximum. For load balancing in wireless ad-hoc networks we propose a design for distributing the load of clusterhead node among under loaded nodes by means of calculating the forwarding Index parameter. By distributing the load of clusterhead we will be able to minimize the average execution time and maximize the lifetime of the overloaded (clusterhead) node.

1.1. RELATED WORK

For the wireless ad-hoc networks, the problem of load balancing is also defined by many authors. We have looked many papers [11, 12, 13, 1, 2] that deal with the problem of load balancing in ad-hoc networks. Reach of these papers attacks the problem of load balancing in different ways. The paper by S.K DAS et. al. [14] tries to devote the load balancing algorithm that would try to find the best node to share the load with while minimizing the communication overhead involved in load – balancing. The another paper by D. Turgut et. al. [2] deals with ad-hoc networks where mobile nodes have been loosely classified into clusters based on their current location. The authors proposed a load – balancing heuristic to extend the battery life of a clusterhead before allowing it to retire and allow another node to become the clusterhead.

2. CLUSTERING MODEL OF WIRELESS AD-HOC NETWORKS

Wireless communication network can be presented by an undirected graph G=(V, E), where V and E are the set of vertices and edges respectively for graph G. Each node $v \in V$ represents a wireless station and every undirected edge $e \in E$ defines a neighbor relationship between two

nodes, that, is to say it indicates two nodes at the end of edge those can communicate with each other, for any nodes u and $v \in V$, if v is an adjacent node of neighbors of u and is not a neighbor of u, we say v is two hop node of u.

Each node has a unique identifier (ID) number and only has a transceiver operated in half-duplex mode. Node requires adjacent nodes and all the topology connection of the whole network through transmitting or receiving default control packet or message. At first every node sends its control packet to indicate its existence. When a node receives a control packet of adjacent node, it updates its related data table. And when it transmits its control packet once again, the packet includes all the information of its adjacent nodes it knew. From this we can see that if all nodes in network transmit a control packet respectively, every node will know all its neighbors. If all nodes transmit another control packet again, the node will know its two hop nodes. With further exchange, every node will know the topology of the wireless network. Based on the locality information, distributed control is implemented.

3. CLUSTERING ALGORITHMS

One of the fundamental problems in wireless mobile computation is efficient cluster formation. It is well known that optimal cluster formation is NP-complete.

In this section we study the problem of distributed cluster formation in a wireless ad-hoc environment. We work on Bluetooth model which is a synchronous system in which every node has a unique ID, but does not know the ID of any other node. A node trying to discover other nodes broadcasts a generic message and does not advertise its ID in the massage. The replying node gives its ID in the reply message, but does not know which node it is replying to. However, after a device has discovered another device, much more information can be passed between them with relatively less overheads. This model is ideally suited to frequency hopping systems, where the devices hop on a sequence of frequencies, and the messages are repeatedly broadcast in order to reach other nodes.

The bluetooth SIG aims to provide solutions for short-range wireless connectivity between pervasive devices, like PDAs, mobile phones, palmtops, laptops, pagers, etc. It is meant to be cable-replacement solution for desktops, keyboards and other peripherals devices. The potential applications range from smart home appliances to wireless connectivity to backbone data networks. Bluetooth is being considered for use by the top players in the consumer electronics market. Products would include wireless headsets, cameras, watches and portable games.

The automotive industry is also looking to use Bluetooth technology as the key solution for onboard wireless communication systems, connecting vehicular and external networks. These and other applications in the office and classroom environments, like shared whiteboards, would make it important for the devices to quickly self-organize into an ad-hoc network.

4. A RANDOMIZED ALGORITHM FOR CLUSTER FORMATION

In this section, closely following Aggarwal et.al.[9], we describe a two-stage algorithm for partitioning the set of nodes into a connected set of star-shaped clusters, while keeping the size of the clusters at their maximum. An important idea used in this algorithm is to make a device continuously broadcast or continuously listen, in order to increase the probability of the message reaching another device.

The first stage of algorithm is randomized, at the end of which each node either becomes a Master-designate or a Slave-designate. For a network of N nodes and maximum cluster size S, the ideal number of Masters is k=[N/(S+1)]. The second stage uses a deterministic algorithm to decide on the final set of Masters and Slaves, and to efficiently assign Slaves to Masters. A Super-master is elected, which is required for counting the actual number of Masters, and for collecting information about all the nodes. This stage also corrects the effects of the randomness introduced in the previous stage. The election of the super-Master is interleaved with the cluster formation, which speeds up the ad-hoc network formation. The super-master can then run any centralized algorithm to form a network of desired topology.

In the following discussion, we use the following terminology:

- Slave-designate: a node which did not succeed in any of the Bernoulli trials, and is not yet part of any cluster.
- Slave: a Slave-designate which becomes part of a cluster.
- Master-designate: a node which had a successful Bernoulli trial, and has not yet collected Inquiry response from enough slave-designates and has not timed out (CLUSTER_TO).
- Master: a node which has collected response from S Slave-designates or has timed out (CLUSTER_TO).
- Proxy-slave: a Slave which has been identified by its Master to participate in the super-master election on its behalf.
- Super-master-designate: a Master which has collected k responses from other cluster, or has reached the SUPERM_TO.
- Super-master: a Master which has response from all other clusters, and has information about all the nodes in the network.

The algorithm given by Aggarwal et.al. [9] is described below:

Stage 1: Each of the N nodes conduct T rounds of a Bernoulli trials with probability of success equal to p. A node which is successful at least once becomes a Master-designate and the remaining nodes become slave-designate.

Stage 2: We make the following additional assumptions on the various timeout value used by the nodes. These timeouts are the same for all the nodes.

Assumption-1: Each node has a CLUSTER_TO value such that if it inquires for this period of time, and there are enough number of nodes in its radio range which are scanning for Inquiry packets, then at least S devices will respond to it.

Assumption - 2: Each node has a SUPERM_TO value such that a node inquiring for this period receives responses from at least 2k nodes that are scanning. For practical purposes, we assume that $P[X > 2k]$ is very small, for reasonably large k.

Assumption-3:A set inquiring nodes catch one scanning node each, well before the SUPERM_TO period.

Algorithm 4.1[1]: Master-designates and slave-designates are using state-1, as described above. Let X be the actual number of master-designated.

Each master-designate inquires continuously until neither the CLUSTER_TO nor the SUPERM_TO is reached. If a response is received from a Slave designate, it made a Slave in its cluster by paging and making a connection to it, as long as the maximum cluster size is not

exceeded. If this is the first Slave of the cluster, the Master-designate instructs it to become a proxy-slave.

When the cluster becomes full, the Master-designate declares itself master, and any future inquiry responses from Slave-designates are ignored. As part of the Super-master election which is interleaved with the cluster formation, the Master/Master-designate collects up to k responses from Proxy-slaves of other clusters, or times out (SUPERM_TO), whichever is earlier. At this point, the node declares itself Super-master-designate.

However , this happens only after CLUSTER_TO has occurred. If the Master-designate has not collected any responses by the CLUSTER-TO period, then it becomes a Slave-designate and starts scanning. Node have unique Ids, and the master with the highest Id is chosen to the Super-master, there is exactly one Super master that is elected. A slave designate continuously scans for Inquiry massages. If, on sending an inquiry response, the inquirer does not page it and establish a connection, then it goes back to scan state. However, If the inquirer connects to it, then it becomes a Slave of the Cluster headed by its inquirer, and stops scanning. If the Master/Master-designate directs it to become a Proxy-slave, it goes into scan for the Super-master election.

5. FORWARDING INDEX AND WIRELESS AD – HOC NETWORKS

Our main objective is to study the problem of evenly distributed cluster formation in ad-hoc wireless environment. It is desirable to have these clusters as evenly distributed as possible over the network to avoid congestion in the network. Clusterhead form a virtual backbone and may be used to route packets (messages) for nodes in their cluster. Nodes are assumed to have non-deterministic mobility pattern. Diffusing node identities along the wireless links forms clusters. Different heuristics employ different policies to elect clusterheads. Several of these policies are biased in favor of some nodes. As a result, these nodes shoulder greater responsibility and may deplete their energy faster causing them to drop out of the network (i. e. there occurs a congestion in the network). Therefore, there is a need to minimize the load of clusterhead. Clusterheads maintain cluster databases for routing purposes.

To avoid the congestion in the network we propose the concept of forwarding index for the clusterhead of the cluster. The clusterhead-forwarding index of the network (cluster) is the minimum value of the largest load occurring at a clusterhead taken over all nodes in the cluster, where load of a clusterhead is defined as the number of paths (routes) passing through that clusterhead. This helps to evenly distribute the responsibility of acting as clusterheads among all nodes to avoid congestion in the network. This congestion is also bad from the fault-tolerance point of view for if clusterhead of such a cluster were to fail a large part of the network come to a halt. Computing forwarding index of general network was shown to be NP-complete [6] by Saad [3] and problem of optimal clustering is also NP-complete.

In a communication network data, messages, etc., are transmitted from each node to any other node. A convenient way to achieve this is to have for every source node a designated route, a sequence of intermediate nodes for every destination. A set of nodes (which are processors or communication centers), with links between some of them for the purposes of communicating data or messages is usually represented by graphs. Generally the nodes are to be interpreted as computer/communication devices. In practice, the networks to be constructed may range from arrays of microcomputers to systems of large geographically remote centers. Instead of speaking of nodes and links we speak of vertices and edges.

The network connecting the n nodes is designed by specifying first the bi-directional communication lines or channels, i.e., those pairs of nodes having direct communication. Interconnection is limited by a port constraint $d \geq 2$ common to each node; i.e., at most d (where d is the degree of the graph) communication lines can be attached to any node. Since it follows in general that not every pair of nodes will have direct communication, the network design must also specify a set of n (n-1) paths called a *routing,* indicating for each x and $y \neq x$ the path or fixed sequence of lines which carries the data transmitted from node x to node y. Implicit here is that in addition to being data sources and sinks, the nodes can serve a forwarding function for the data being communicated between other nodes. Note that, generally, the path from node x to node y need not be the reverse of the path from node y to node x.

If some nodes or links fail, it is important to know which paths of the network are destroyed, and quite naturally it seems a 'good' routing should not load any vertex or edge too much, in the sense that not too many paths of the routing should go through it. In order to measure the load of a vertex, Chung, Coffman, Reiman and Simon introduced in [2] the notion of Forwarding Index [2,3,10].

The network *forwarding index* ξ, defined as the maximum number of paths passing through any node, i.e., ξ is the maximum forwarding being done in the network. With n and d given we consider the specific problem of finding networks that minimize the forwarding index; we call this *forwarding index problem.*

Fig. 1 shows an example for n = 6 and d = 3. According to the routing indicated, nodes 1 and 4 forward the traffic on one path each; nodes 5 and 6 forward the traffic on two paths each; and nodes 2 and 3 forward the traffic on a total of four paths each. Thus $\xi = 4$ for this network.

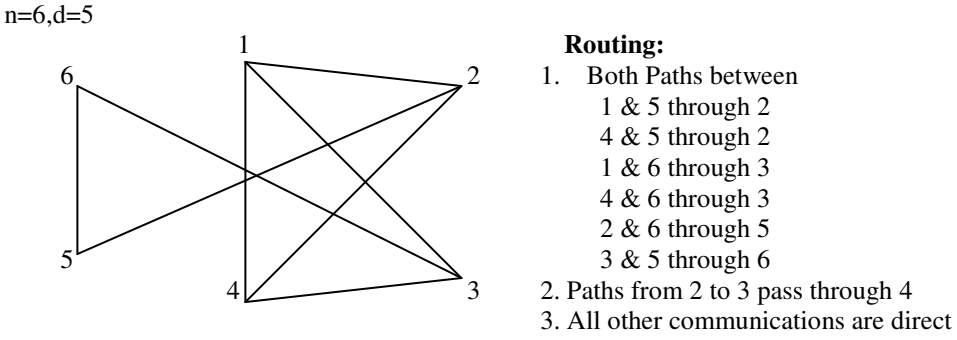

Figure 1.

Concrete applications of the forwarding index problem can be found in problems of maximizing network capacity. For example, assume symmetric transmission requirements in the sense that the transmission rate, say λ, is the same for each node to every other node. The total rate at which data originates and terminates at each node is, therefore $2(n\text{-}1)\ \lambda$, and the total transmission rate among the nodes is $n\ (n\text{-}1)\ \lambda$.The amount of forwarding at a node is assumed to be limited by a capacity c common to all nodes. Specifically, the local transmission rate at a node $2(n\text{-}1)\ \lambda$, plus the rate at which it forwards data for other nodes cannot exceed c. In Fig. 1, for example, since the nodes 2 and 3 forwards the most traffic and since the traffic at these nodes is $2(n\text{-}1)\ \lambda + 4\lambda = 14\lambda$, we must have $c \geq 14\lambda$. The constraint on node capacity

requires that $(2n\text{-}1)\,\lambda + \xi\,\lambda \le c$. The local traffic originating or terminating at each node must, therefore, satisfy

$$(2n\text{-}1)\,\lambda \le 2(n\text{-}1)c/\,\xi + 2(n\text{-}1) \qquad \ldots\ldots\ldots(1)$$

thus defining an *effective* node capacity $c/\,\xi + 2(n\text{-}1)$. The corresponding bound on the total data transmission rate defines the network capacity

$$n(n\text{-}1)\,\lambda \le (nc/2)/(1 + \xi/2(n\text{-}1)\,) \qquad \ldots\ldots\ldots(2)$$

In Fig. 1 the effective node capacity is $5c/7$ and the network capacity is $15c/7$. From (1) and (2) the problem of maximizing capacity for given n and d clearly reduces to the forwarding index problem.

5.1. NOTATION AND TERMINOLOGY

More formally, let G=(V, E) denote a network with vertex-set V(G) and edge-set E(G). If x,y are vertices in G, then a route is a path between x and y, denoted by $R(x, y)$. A routing R in G (graph G has n vertices) is defined as a set of $n(n\text{-}1)$ routes specified for all ordered pairs of vertices of G, one route for each ordered pair.

Let us call the *load of a vertex x* in a given routing R of a graph G, denoted by $\xi(G, R, x)$, the number of paths of R going through x (where x is not an end vertex). The vertex forwarding index of a network (G, R) is the maximum number of paths of R going through any vertex x in G and is denoted by $\xi(G, R)$,

$$\xi(G, R) = \max_{x \in V(G)} \xi(G, R, x)$$

The minimum forwarding index over all possible routings of a graph G will be denoted by $\xi(G)$ and be called the *vertex-forwarding index of* G. The minimum taken over all the routings of shortest paths will be denoted by $\xi_m(G)$.

$$\xi(G) = \min_{R} \xi(G, R) \quad \text{and} \quad \xi_m(G) = \min_{R_m} \xi(G, R_m)$$

Since the notion of load in networks (always limited in practice by the capacity) is at least as important for links as for nodes, it is interesting to introduce and study the same concepts for the edges of a graph.

Therefore we define the *load of an edge e* in a given routing R of G as the number of paths of R, which go through it, and denote it by $\pi(G, R, e)$. Then the edge forwarding index of (G, R), denoted by $\pi(G, R)$, is the maximum number of paths of R going through any edge of G

$$\pi(G, R) = \max_{e \in E(G)} \pi(G, R, e)$$

and the edge-forwarding index of G is defined as

$$\pi(G) = \min_{R} \pi(G, R) \quad \text{and} \quad \pi_m(G) = \min_{R_m} \pi(G, R_m)$$

Clearly $\xi(G) \leq \xi_m(G)$ and $\pi(G) \leq \pi_m(G)$. The equality however does not always hold as can be seen in the following example. The forwarding index of a cluster for shortest path routings is $O(n^2)$ and this is best possible.

Example: Let W_6 be the wheel on 7 vertices, with vertices 0,1,2,3,4,5 on a cycle and a vertex c joined to all the previous ones. Let us define routing of shortest paths R_m in W_6 as follows: for every i, $0 \leq i \leq 5$, $R_m(i, i+2) = R_m(i+2, i) = R_m(i, i+1, i+2)$ (where the vertices are taken modulo 6), and for $0 \leq i \leq 2$, $R_m(i, i+3) = R_m(i+3, i) = R_m(i, c, i+3)$. We have $\xi(W6, R_m, c) = 6$ and for any i, $0 \leq i \leq 5$, $\xi(W_6, R_m, i) = 2$, and clearly $\xi_m(W_6) = 6$. Also for any i, $\pi(W_6, R, ic) = 4$ and $\pi(W_6, R_m, i\ i+1) = 6$ and clearly $\pi_m(W_6) = 6$.

6. CONCLUDING REMARKS AND DETAILS FOR FURTHER STUDY

In this paper we have studied a very fundamental problem of how several nodes organize themselves into an ad-hoc network. We study heuristics for cluster formation and observe that the resultant work has the best possible forwarding index asymptotically. The randomized algorithm for cluster formation can be slightly altered to yield very good forwarding index. This algorithm has many applications, the foremost is to scatternet Bluetooth. According to Bluetooth specification, the smallest network unit is a piconet, consisting of a device and several Slave Bluetooth devices.

Bluetooth devices 'discover' each other by executing the Inquiry and Page procedures. In the Randomized algorithm, continuous Inquiry and inquiry scan is used. It is clear that an inquiry procedure can take a fair amount of time even for two devices to discover each other. Once a connection is established, any amount of information can be exchanged between the nodes without much overhead. The proposed study of forwarding index is very useful for load balancing by means of minimizing the load of the clusterhead (Master) node and this maximizes the life time of the clusterhead node.

The proposed study in our paper can be further extend to explore the broadcasting properties i.e. broadcasting radius. For good quality of clustering algorithms the low broadcasting radius can be one of the important parameter.

REFERENCES

[1] Turgut D, Turgut B., Das S. K., Elmasri R,(2003), "Balancing Loads in mobile adhoc networks, Telecommunications, ICT 2003, 10th International Conference on, Volume:1, 23 Feb.1-March 2003.

[2] F.Chung, E.Coffman, M. Reiman, and B. Simon, (1987), "The Forwarding Index of communication network," IEEE Transaction Info. Theory, 33(1987), pp. 224-232.

[3] R. Saad, (1993), "Complexity of the Forwarding Index problem," LRI technical Report & SIAMS, Disc. Math., 6 (3), pp. 418-427, to appear.

[4] A. Shastri, (1998), "Forwarding index and connectivity of Communication Networks," Proceedings of IEEE International Conference on Networking Indian and the World, Ahmedabad, pp 71-77.

[5] C-C Chiang, (1997), "Routing in clustered multihop, mobile wireless network with fading channel," Proc. IEEE SICON' 97, pp. 197-211.

[6] M. gary, Micheal R., and Johnson, David S.; Freeman, (1979), "*Computers and Interactablity:" A Guide to he theory of NP-ompleteness* NewYork.

[7] Y. Manoussakis and Zs. Tuza;, (1989), "The Forwarding Index of directed networks," technical report L.R.I. No. 482, University of Paris-XI; Disc. Appl. Mathematics, 68, 1996, pp. 279-291. to appear.

[8] Y. Manoussakis and Zs. Tuza;, "Optimal routings in communication networks with linearly bounded forwarding index," networks, to appear.

[9] A. Aggarwal et. al., (2001), "Clustering Algorithms for Wireless Ad-hoc Networks,", preprint.

[10] A. Shastri, (2001), "Wireless Ad-Hoc Networks with good Broadcasting and Load Balancing Properties," ATM Interact Symp. On Broadband Networking in the New millennium, New Delhi, India, pp. 43-50.

[11] S.K. Das, S.K. Sen and R. Jayaram, (1998), **"**A Novel load Balancing Scheme for the Tele-Traffic Hot speed Problem in Cellular Networks"**,** ACM/Baltzer Journal on Wireless Networks, Vol. 4, No. 4, pp. 325-340.

[12] S.K. Das, S.K. Sen and R. Jayaram, and P. Agarwal,(1997), "A distributed Load Balancing Algorithm for the Hot cell Problem in Cellular Mobile Networks", Proc. of Sixth IEEE International Symposium on High Performance Distributed Computing, Portland, Oregon, pp 254-263.

[13] R. Prakash, A. D. Amis**,** (2000), **"**Load Balancing in Wireless ad-hoc networks"**,** Application Specific Systems and Software Engineering Technology, Proc. 3rd IEEE Symphosium, 24-25.

Authors

Prof. Aditya Shastri is presently Professor of Computer Science & Vice Chancellor at Banasthali University. He received his M.Sc. (Tech.) Computer Science & M.Sc. (Hons.) Maths degree from BITS-Pilani, India before completing his Ph.D. at Massachusetts Institute of Technology, Cambridge USA in 1990. His research interests are in Discrete Mathematics, Graph Theory and Mobile Computing. He has written more than 50 research papers in journals of repute and authored 5 textbooks. Of late, he is engaged in the development of Banasthali University as its Chief Executive Officer and Chief Academic Officer.

Dr. Reena Dadhich is presently working as a Associate Professor and Head of the Department of Master of Computer Applications at Engineering College Ajmer, India. She received her Ph.D. (Computer Sc.) and M.Sc. (Computer Sc.) degree from Banasthali University. Her research interests are Algorithm Analysis & Design and Wireless Ad-Hoc Networks. She has more than 11 years of teaching experience. She has written many research papers.