Experimental Evaluation of Scalability and Reliability of a Multicast Protocol for MANETs

Safiullah Faizullah
Rutgers University
Piscataway, NJ, USA

Arshad Shaikh
Isra university
Hyderabad, Pakistan

ABSTRACT
In this era of social media, real time video multicast over Mobile Ad-hoc Networks (MANETs) is subject of active research and new products are in use as a result of these research advancements. We have earlier proposed a simple and novel approach for multicasting in MANETs that is particularly suited for multicasting live video/audio streams, which gives efficient solution to this problem. The proposed solution is lightweight, scalable and is general that it can be made to work with any underlying unicast routing protocol. Experimental evaluation of scalability and reliability aspects of our solution is presented in this paper.

Keywords
MANET, Mobile Ad-hoc Networks, Scalable, Multicast Tree Management, Multicast Tree Maintenance, Lightweight, Multicasting.

1. INTRODUCTION
With the advancements in networked applications that multicast (such as audio, video and teleconferencing) and with increased requirement for more network resources (such as bandwidth), there is a need for better ways to deliver data so that the network resources are efficiently utilized. In the era of social networks, Mobile Ad-Hoc Networks (MANETs) are an essential link in this delivery.

Multicast connections are connections between one or more senders and a number of members of a group which suites very well to these applications. As the main aim of multicasting is to be able to send data from a sender to the members of a group in an efficient manner, this mode of communication is important due to the increased use of new point to multipoint applications, like web-based e-learning, web-based radio and TV, other collaborative environments, and movies/video on-demand video. Extensions of multicasting include videoconferencing and teleconferencing applications where each node in the group can be a sender as well. These applications may use separate trees for each sender or may utilize a common/shared tree. While former requires more resources in the routers to maintain multiple trees, the later may result in longer delivery paths and hence consume more network resources resulting in decreased throughput and increased delay.

In scalable Multicasting, where by adopting Multicast group addressing in the Internet help eliminate the need for the source to know the identity of all the receivers. In this paradigm, packets are delivered to each recipient who has declared its membership in the multicast group. The routers, using one of many algorithms, determine an optimal distribution tree spanning each recipient node and forward data packets along this tree. Moreover, the routers at the branching vertices automatically create copies of the data packets and send them along each branch. Also, since nodes are allowed to join or leave multicast sessions, this results in a dynamic distribution tree [1, 1a]. The current best-effort nature of Internet is posing many challenges for multicasting real-time video streams [3] or other near-real time multicasting [Safi]. Mobile Ad-hoc Networks (MANETs) which are infrastructure-less collection of mobile nodes communicating over wireless link (a short range CSMA/CA transceiver) with nodes in their range is drastically different that the wired networks. In this network, each node in a MANET agrees to act as a router for other nodes, i.e., forwards their packets, hence establishing a multi-hop end to end communication network [4]. With the flexibility that MANETs provide, multicasting is more challenging due to node mobility and frequent changes in topology. In this case, the multicast distribution structure, represented as tree, needs to be updated continuously and hence the protocols designed for multicasting in wired networks do not typically perform well in MANETs as well. A point to be noted is that the multicast routing protocols for MANETs can exploit the inherent wireless broadcast available locally at each node to avoid making explicit copies of multicast data packets. This enables that the branching decision to be made in a distributed fashion contrary to the wired networks where each router needs to make explicit copies to create branches.

To the best of knowledge, several protocols for multicasting in MANETs have been proposed and studied in literature; none seems to make use of the local broadcast property of MANETs [references or names].

The authors of this work have proposed a Simple, Lightweight and Intuitive Multicast protocol called "SLIM" [2]. The proposed protocol is highly scalable and as reliable as the scheme that utilizes multiple separate unicast connections from source to the destinations. SLIM is independent of the underlying unicast protocol yet it makes use of the underlying unicast protocol to determine paths between source and the destination nodes, thus it showed that our approach is very promising. Now we study our scheme for scalability and reliability and present the results of simulations conducted in this paper.

The rest of the paper is organized as follows: next section (Section II) presents background and gives a short literature review for MANETs. Section III highlights the proposed SLIM protocol. Section IV presents the results of simulations and Section V presents the conclusion, highlighting some future research directions, which is followed by references.

2. BACKGROUND LITERATURE REVIEW
In the literature, we find many proposed solutions to the problem of multicasting in general [1] and for MANETs [2-11].

The authors of this work have proposed a Simple, Lightweight and Intuitive Multicast protocol called "SLIM" [2]. We have reviewed and highlighted the shortcomings of the prominent strategies in our earlier work— thus, motivating our original work presented in [2] which showed that our approach is very promising. Now we study our scheme for scalability and reliability and present the results of simulations conducted in this paper. An extensive literature review was provided in in [2] with a brief summary provided in this section.

A simple approach for providing multicast in a MANETs is flooding in which each node in the network receives all the data packets. Although this approach may be advised [3] to achieve a reliable multicast in a highly dynamic network, its drawback of having a high overhead is obvious.

A classification of MANET multicast protocols in [5] is provided by Corderio and Agarwal who have classified the them into four categories based on how route to the group members is created, viz. Tree-based, Meshed-Based, Stateless, and Hybrid protocols [5].

Ad-hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS) [6] is an on-demand multicast routing protocol, which facilitates multiple senders by constructing a shared multicast tree. The tree is rooted at a special node Sid (mostly the first sender of the multicast session) and spans all the nodes in the network. The actual delivery tree is formed dynamically as a sub-tree of this spanning-tree depending upon the set of nodes interested in receiving the multicast data. AMRIS dynamically assigns an (non-unique) id-number to each node in the network depending upon its distance from the root node. The ordering between id-numbers is used to direct the multicast flow. These id-numbers help the nodes dynamically leave and join a session, as well as adapt rapidly to changes in link connectivity. In the initialization phase the Sid announces the availability of multicast session by flooding NEW-SESSION message. Each node in the network upon receiving a New-Session message computes its id-number and determines its potential parent in the multicast tree, and then further propagates the NEW-SESSION message to its neighbors. Any node interested in receiving the multicast sends JOIN-REQ message to its parent in the spanning-tree. If the parent node is already a part of the delivery sub-tree, it acknowledges with a JOIN-ACK message, otherwise it joins the delivery sub-tree in a recursive manner. AMRIS employs a beaconing mechanism to detect link failures and defines branch reconstruction procedures to handle such situations. However there is a high possibility of packet drop until the broken link is detoured through these procedures. From their paper it is not clear how the delivery tree will be pruned if a node leaves the multicast session.

Multicast operation of Ad hoc On-Demand Distance Vector routing protocol (MAODV) [7] is a direct extension of unicast AODV. It uses the flooding mechanism of AODV to construct the multicast tree. It creates bi-directional shared multicast trees connecting multicast sources and receivers. A node that wishes to join a multicast group (either as a sender or as a receiver) originates an RREQ message. Only a member of the desired multicast group may respond to this RREQ. Each multicast group has a group leader whose responsibility is to maintain the group sequence number, which is used to ensure freshness of routing information. Periodic HELLO messages are sent by the group leader and help detect any broken links. The downstream node of a broken link starts the repair process by broadcasting RREQ with a TTL equal to the hop count to the group leader. If any tree node receives an RREQ with TTL value larger than the hop count to the group leader, the tree node replies the broken node with RREP that the repair is successful. The main drawbacks of MAODV are long delays and low packet delivery ratios which are due to broken links in situations of high mobility and heavy traffic load.

While tree distribution structure has the advantage of being loop free, it is subject to an entire tree reconfiguration even with a single link failure. Mesh based protocols provide multiple (redundant) paths between any source and destination nodes and hence are more reliable and tolerant to link failures [5].

On-demand Multicast Routing Protocol (ODMRP) [8] is a mesh based protocol, which employs a subset of nodes to forward the multicast packets. A soft state approach is taken to maintain group membership. Multicast source periodically broadcasts a Join-Query (JQ) control packet to the entire network to refresh the membership information and updates routes. An intermediate node may receive multiple JQ packets. After validating the TTL and avoiding duplicates, the intermediate node stores in its routing table the node ids of multiple parent nodes which can supply the multicast. A destination node may also receive multiple JQ packets. After TTL validation and duplicates removal, the destination node creates a Join-Reply (JR) packet containing a list of potential parent nodes and broadcasts it to all the neighbors. When an intermediate node receives a JR packet, it checks if its own id is listed as one of the potential parents. If so, it sets its FG_FLAG (Forwarding Group Flag) and broadcasts it own JR Packet to its neighbors. Join-Reply is propagated until it reaches the multicast source. After establishing a forwarding group, the source multicasts data packets to receivers via selected routes. Upon receiving multicast data packet, an intermediate node forwards it only when it is not a duplicate and the node's FG_FLAG has not expired. No explicit control packets are needed to join or leave the group. If a multicast source wants to leave the group, it simply stops sending JQ packets. Similarly a receiver can stop replying with JR packets in order to leave. Nodes in the forwarding group are demoted to non-forwarding nodes if not refreshed before timeout.

Forwarding Group Multicast Protocol (FGMP) [9] can be viewed as a limited-scope flooding, a flooding within a selected forwarding group (FG), using a virtual mesh of point-to-point unicast routes. Each node in FG forwards data packets if the forwarding flag is set and the timer is not expired. FGMP describes two approaches to elect and maintain the forwarding group: FGMP-RA (Receiver Advertising) and FGMP-SA (Sender Advertising).

FGMP is similar method to ODMRP, whereas their main difference is the way group meshes are established. Both FGMP and ODMRP suffer from scalability problems due to flooding of control packets.

Tree-based as well as mesh-based approaches have an associated overhead of creating and maintaining the delivery structure. In a highly dynamic MANET environment, this overhead of maintaining the delivery tree/mesh increases considerably. Stateless multicast is proposed wherein a source explicitly mentions the list of destinations in the packet header. Stateless multicast approaches focus on small group multicast and assumes the underlying routing protocol to take care of forwarding the packet to the respective destinations based on the addresses contained in the header.

The tree-based protocols provide better throughput but little reliability, whereas the mesh-based protocols provide a much robust multicast at the cost of increased network load. AMRoute [10], MCEDAR [11] and MHMR [12] are some hybrid protocols that are proposed to brew the advantages of both mechanisms.

3. THE PROPOSED SLIM ALGORITHM

In our approach, each intermediate node commits to relay the multicast packets in its antenna range provided that there are listeners (subscribers or other intermediate nodes) interested in receiving the stream through them. This state is kept using a single flag per active multicast stream in each router node. Nodes interested in receiving the multicast transmission periodically (say every T seconds) send MTREQ (multicast transmission request) message towards the source using ordinary unicast mechanism (AODV, DSDV, etc.) which is currently employed by the network. All the intermediate nodes (including the sender) in the path of this message agree to relay the multicast stream for the next (T+D) seconds, where D is a cushion time sufficient enough for the dependent subscribers to re-express their interest. Clearly this defines a dynamic multicast tree. A node being an intermediate router for more than one subscriber commits to relay for T+D seconds from the last MTREQ received from any of the subscribers. Hence a single entry/flag is needed in the routing table irrespective of the number of dependant subscribers or branching. An intermediate node which is no longer in the path of any active subscriber automatically stops relaying the stream after the expiry of T+D commitment interval.

Consider, for example, the scenario presented in Figure 1, in which node 5 is the source of a live stream and node 13 is interested in receiving the stream. Using the inherent unicast methodology (AODV, DSDV, etc.), node 13 sends an MTREQ message to node 5. Suppose this message takes the path 13->10->7->5. Each of the nodes in this path (i.e., nodes 10, 7 and 5) sets a flag to record their commitment to relay the multicast traffic for next 2 seconds (assuming T=D=1). To fulfill this commitment, node 5 starts transmitting packets in its area of coverage. Nodes 2, 3, 6, 7 and 9 being its neighbors listen to this transmission however only node 7 is committed to repeat the packets into its area of coverage. Similarly, node 10 repeats the packets coming from node 7 and hence the packet is received by node 13. Node 13 will keep sending MTREQ packets to node 5 after every 1 second (T=1) to reset the flag timers of intermediate nodes and keep them committed.

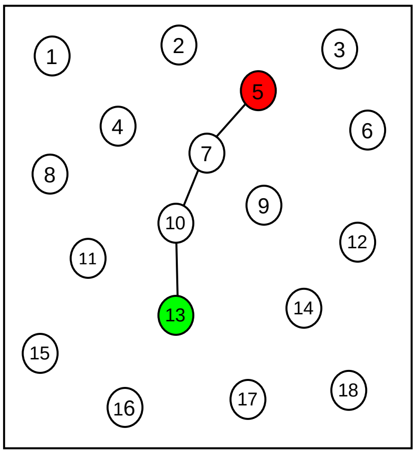

Figure 1: A multicast scenario

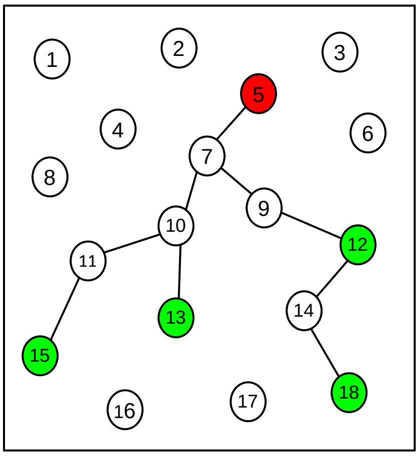

Figure 2: Multicast scenario after some time

After some time suppose nodes 18, 12 and 15 also get interested in receiving the multicast data. These subscriber nodes will also start periodic MTREQ packets towards node 5, hence creating the multicast commitment tree as shown in Figure 2. With nodes 9 and 10 committed in the neighborhood of node 7, any packet relayed by node 7 will get repeated by these two nodes and as such node 7 won't have to make explicit copies of data packets for 9 and 10.

Further, it can be noted that node 12 behaves both as router and receiver for this multicast and hence not only consumes the traffic but also relays it for subsequent subscribers.

In case of any change in topology (for example, due to the mobility of any subscriber or any intermediate node) the subsequent unicast of MTREQ will re-determine the tree. If any intermediate node doesn't receive any MTREQ packet within the expiry of commitment timer, the node clears the flag and stops relaying the multicast packets.

The proposed scheme is ideally suited for multicasting live streams and is lightweight in the sense that the intermediate nodes (routers) do not have to maintain the list of subscribers receiving the transmission through them. Neither do they have to multiply the multicast traffic (send multiple copies) in case of a branch.

4. EXPERIMENTAL METHODOLOGY AND RESULTS

In this section, we detail the experimentation methodology that we adopted for this work. Both the traditional algorithms and our algorithm were incorporated into an off-the-shelve network simulator tool that we will simply refer to as NST in this paper. We needed to add some missing functionalities to the simulator.

To conduct the simulation studies, we will be using randomly generated networks of varying complexities and sizes to establish the initial scenario and setup. This ensures that the simulation results are independent of the characteristics of any particular network topology. Using randomly generated network topologies as initial test bed also provide the necessary flexibility to tune various network parameters to study the effect of these parameters on the performance of the technique. To generate random graphs, we will use a method similar to [10]. A 25-100 nodes ad hoc network was simulated and multicast group comprising of 20-50 members with 5-10 senders.

Node mobility and wireless functionality was simulated using the random waypoint with each node is stationary for x seconds. The node then moves towards some destination randomly picked in a randomly generated grid. Sparse graphs, containing a small percentage of the total number of possible nodes, with low average group members are more representatives of real networks and pose a tougher problem. To generate node addition and node deletion requests to multicast groups, after the initial simulation network scenario setup, we will employ probabilistic model similar to [10] that allows control of the relative frequencies of add and delete requests.

The results presented in his paper are based on averages for 10-20 iterations of same network setup parameters (initial topology, traffic, groups, etc.) in order to reduce the effect of specific network configuration and node mobility patterns on the overall simulation results. Implementation details including overhead for multicast protocols are removed from results, the effect of which will be studied in future. We also impose a bound on the number of sub-flows that we create. Lastly, we conducted large number of experiments with varying load.

Figure 3 shows the end-to-end delay in seconds (y-axes) for varying load (x-axes), representing the number of receivers, conducted on several environments. It is apparent that the SLIM algorithm scales well and the delay is in acceptable range for quality real-time video delivery. We can note that under heavy loads, the algorithm suffers some delays as the load increases.

As the load on the network is increased, some new sessions and members to established groups may not be admitted, this is quantified in Fig. 4 and Fig. 5. Fig. 4 shows the number of new sessions not admitted (y-axes) for varying load (x-axes), representing the number of receivers, conducted on several environments. Fig. 5, depicts the same data as percentage of new sessions not admitted. It is apparent that the SLIM algorithm is capable of handling load well and the number of sessions not admitted (which increases with load) is bounded. In Fig. 5, we can see that apparat from the case with low load, where the impact of the load doesn't seem significant for both various network scenarios, the algorithm scales well for various network scenarios that were subjected to different load models. We can see that the result show that the new sessions denial is bounded and within acceptable range for quality real-time video delivery. We can note that even under heavy loads, the new sessions not admitted are below 5% as worst case scenario. One element of surprise was that the algorithm did well in dense networks even those which were large in terms of nodes as opposed to moderate networks when subjected to similar heavy loads. However, this difference was within 1% as worst case scenario.

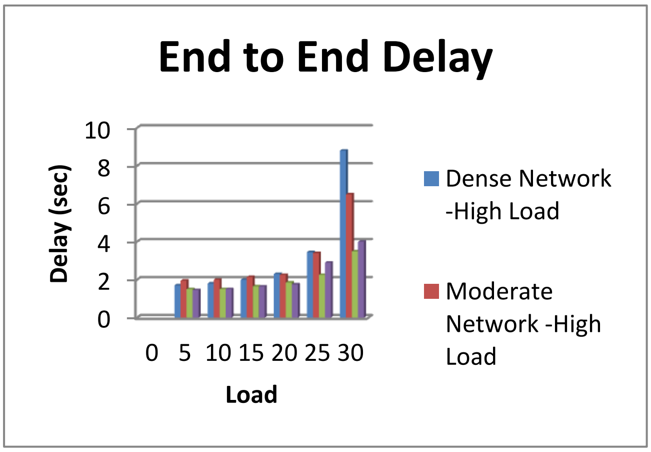

Figure 3: End to End Delay

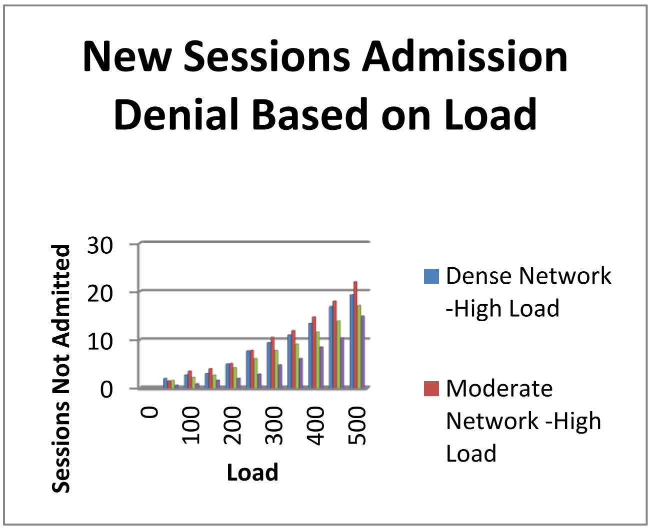

Figure 4: New Sessions Admission Denial as Function of Load

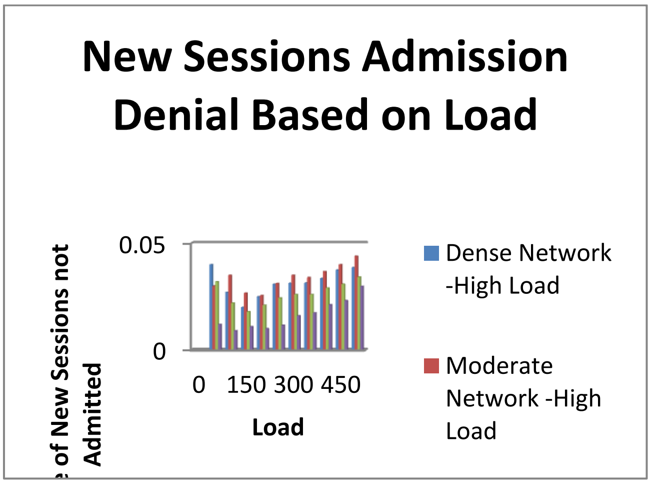

Figure 4: Percentage of New Sessions Admission Denial as Function of Load

5. CONCLUSION AND FUTURE DIRECTIONS

SLIM, a novel approach for creating and maintaining multicast trees in MANETs was presented in our earlier work [2]. The approach is generic in the sense that it can be used with any underlying routing protocol. The performance evaluation that we conducted in this paper shows that the algorithm is promising with high delivery ratio and low overheads. We would like to extend this experimental evaluation to other aspect to ensure that the protocol is robust. We plan that in near future we will conduct further experimental evaluation of the proposed protocol using various unicast protocols, studying the scalability and reliability of the protocol, and comparison to other schemes.

6. REFERENCES

[1] S. Faizullah, A. Shaikh, "An Innovative and Efficient Multipath Multicast Mechanism for Data Streams", in the proceedings of The IASTED International Conference PDCN 2014, February 2014 Innsbruck, Austria.

[2] A. Shaikh, S. Faizullah, et. al., "SLIM - Simple Lightweight and Intuitive Multicast Protocol for MANETs", International Journal of Computer Applications (IJCA), Vol.94, No. 3, May. 2014, pp 45-47.

[3] X. Li, M. H. Ammar, S. Paul, "Video Multicast over the Internet", IEEE Network (Volume:13 , Issue: 2), pp. 46-60, March/April 1999.

[4] A. Shaikh, D. Vasan, K. H. Mohammadani, “Performance Analysis of MANET Routing Protocols– A Comparative Study”, International Journal of Computer Applications, Vol.83, No.7, Dec. 2013.

[5] C. de Morais Cordeiro, D. P. Agrawal, “Ad hoc and sensor networks: theory and applications”, World Scientific, 2011.

[6] C. W. Wu, Y. C. Tay, "AMRIS: A multicast protocol for ad hoc wireless networks" In Military Communications Conference Proceedings 1999, Vol. 1, pp. 25-29, MILCOM 1999 IEEE, 1999.

[7] E. M. Royer and C. E. Perkins, “Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol” ACM / IEEE international conference on mobile computing and networking 1999, pp. 207-218, Aug. 1999.

[8] S. Lee, W. Su, and M. Gerla, “On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks,” ACM/Baltzer Mobile Networks and Applications, Vol. 7, Issue 6, pp. 441-453, Dec. 2002.

[9] C.-C. Chiang, M. Gerla, and L. Zhang, "Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks", ACM-Baltzer Journal of Cluster Computing: Special Issue on Mobile Computing, Vol. 1, No. 2, 1998.

[10] R. Talpade, A. McAuley, M. Liu, E. Bommaiah, "AMRoute: Adhoc Multicast Routing Protocol" In ATIRP Conference Proceedings. 1999.

[11] P. Sinha, R. Sivakumar, V. Bharghavan, "MCEDAR: multicast core-extraction distributed ad hoc routing" In Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE, pp. 1313-1317, IEEE, 1999.

[12] B. An, S. Papavassiliou, "MHMR: mobility-based hybrid multicast routing protocol in mobile ad hoc wireless networks" Wireless Communications and Mobile Computing 3, no. 2 (2003): pp. 255-270.

7. AUTHOR’S PROFILE

Safi Faizullah received his Ph.D. in Computer Science from Rutgers University, New Brunswick, New Jersey, USA in 2002. He also received MS and M. Phil. Degrees in Computer Science from Rutgers University, New Brunswick, New Jersey, USA in 2000 and 2001, respectively. Dr. Faizullah also earned his BS and MS degrees in Information and Computer Science from KFUPM, Dhahran, KSA in 1991 and 1994, respectively. His research interests are in computer networks, mobile computing, wireless networks, distributed and enterprise systems. He has authored over twenty refereed journals and conference papers. Dr. Faizullah works for Hewlett-Packard and he is a Visiting Scholar/Adjunct Professor of Computer Science at Rutgers University. He is a member of IEEE, SCIEI, PMI and ACM.

Arshad M. Shaikh received his MS degree in Computer Science from Rutgers University, New Brunswick, New Jersey, USA in 2002. He also received MS and BS from National University, Karachi, Pakistan. His research interests are in programing languages and computer networks. He has authored several conference papers. Mr. Shaikh worked for National University, Karachi, Virtual University, and Isra University, Hyderabad, Pakistan as lecturer.