Packet Switched Networks for Communications within Large Multi-Core Systems on Chip

Rakhi Mutha

***Abstract*—Packet-switched networks for communications within large multi core systems on-chip are made for enhanced performance, scalability, modularity, and design productivity more than previous communication architectures such as busses and dedicated signal wires. Multiprocessor architectures and platforms have been introduced to extend the applicability of Moore's law. They depend on concurrency and synchronization in both software and hardware to enhance the design productivity and system performance. These platforms will also have to incorporate highly scalable, reusable, predictable, cost- and energy efficient architectures. With the rapidly approaching billion transistors era, some of the main problems in deep sub-micron technologies which are characterized by gate lengths in the range of 60-90 nm, will arise from non-scalable wire delays, errors in signal integrity and unsynchronized communications. These problems may be overcome by the use of Network on Chip architecture. The proposed paper is a survey of various research papers and contributions in NOC area.**

***Index Terms*—Packet-switched networks, multi-core, on chip architectures, network on chip.**

I. Introduction

Sending a global signal across the chip within real-time working is typical to perform [1]. The circuit will be better for Electro Magnetic Interference; If System on Chip is synchronized by a global clock signal [2]. The normal system designs are generally depends on critical paths and clock trees. These critical paths and clock trees contribute to an increased amount of power consumption so that, SoCs are not power efficient. But, these clock trees difficult to manage due to clock skew problems [3]. As compared-to synchronous designs, asynchronous designs are modular and do not suffer from issues such as clock skew, higher power consumption and EMI. However, designing asynchronous systems is a more complex task as compared to designing a synchronous system. In the case of an asynchronous system designing a glitch free circuit and managing clock arrival time are complicated in the case of an asynchronous system. There is not much support from the EDA (Electronic Design Automation) industry for asynchronous systems. Thus, researchers have combined the ideas of synchronous and asynchronous designs. One such strategy is globally asynchronous and locally synchronous solution; it divides a system into smaller, locally decoupled synchronous regions and then composes a few of them to yield a localized sub system. These synchronous regions and subsystems would be easier to integrate into a global solution and verify. There will be an asynchronous way in which all the local synchronous regions will communicate at the system level. Therefore, these different synchronous regions need not have to be synchronized to a single global clock. This approach will reduce the requirement for chip-wide clock trees; the designers could focus on local synchronous regions only, which would be far less complex than the complete system. Since one has the flexibility to reduce the clock speed of a given synchronous region (or node) independent of other such regions, the amount of power consumption in a system can be managed better and reduced. NOC can improve design productivity by supporting modularity and reuse of complex cores. Thus, it enables a higher level of abstraction in the architectural modeling of future systems.

Manuscript received May 23, 2012; revised June 24, 2012.

R. Mutha was with the JIET Group of Institutions, Jodhpur, Rajasthan , India. (e-mail: rakhimutha@gmail.com).

II. Topological Survey

There are various topologies used for NOC from the communication perspective. Like mesh, torus, ring, butterfly, octagon and irregular interconnection networks [4]-[6]. Researchers have used these topologies for NOC implementations. Kim have used a star-based NOC that communicated using the principle of CDMA (Code Division Multiple Access) [7]. Adriahantenaina proposed a tree-based implementation of NOC where each node in the tree behaves as a router in NOC[8]. Pande compared various network topologies for interconnection networks in terms of latency, throughput, and energy dissipation. Several researchers have suggested that 2-D mesh architecture for NOC will be more efficient in terms of latency, power consumption and ease of implementation, as compared to other topologies. The Octagon NOC demonstrated in is an example of a novel regular NOC topology [9].

III. Router Architecture

Packet-switched networks implemented in NOC architectures. This has led to new and efficient principles for design of routers for NOC [10]. Assume that a router for the mesh topology has four inputs and four outputs from/to other routers, and another input and output from/to the Network Interface (NI). Routers can implement various functionalities from simple switching to intelligent routing. Since embedded systems are constrained in area and power consumption, but still need high data rates, routers must be designed with hardware usage in mind. For circuit-switched networks, routers may be designed with no queuing (buffering). For packet-switched networks, some amount of buffering is

needed, to support burst data transfers. Such data originate in multimedia applications such as video streaming. Buffers can be provided at the input, at the output, or at both input and output [11]. Various designs and implementations of router architectures based on different routing strategies have been proposed in the literature. Wolkotte proposed circuit switched router architecture for NOC [12]. Dally and Towles proposed a packet switched router architecture [13]. Albenes and Frederico provided a wormholebased packet forwarding design for a NOC switch [14].

IV. Routing Algorithm Survey

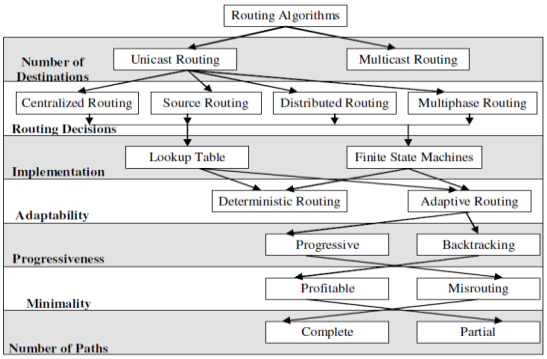

Fig. 1. Routing protocols.

Routing algorithms can be classified in various ways, as shown in Fig. 1. In unicast routing, the packets have a single destination, while in the case of multicast routing, the packets have multiple destinations. For on-chip communication, unicast routing strategies seem to be a practical approach due to the presence of point-to-point communication links among various components inside a chip. Based on the routing decision, unicast routing can be further classified into four classes: centralized routing, source routing, distributed routing and multiphase routing. In centralized routing, a centralized controller controls the data flow in a system. In case of source routing, the routing decisions are taken at the point of data generation, while in distributed routing, the routing decisions are determined as the packets/flits flow through the network. The hybrid of the two schemes, source and destination routing, is called multiphase routing. Routing algorithms can also be defined based on their implementation: lookup table and Finite State Machine (FSM). Lookup table routing algorithms are more popular in implementation. They are implemented in software, where a lookup table is stored in every node. We can change the routing algorithm by replacing the entries of the lookup table. FSM based routing algorithms may be implemented either in software or in hardware. These routing algorithms may further be classified based on their adaptability. Deterministic routing always follows a deterministic path on the network. Examples of such routing algorithms are XY routing, north first, South first, East first, and west first. Adaptive routing algorithms need more information about the network to avoid congested paths in the network. These routing algorithms are obviously more complex to implement, thus, are more expensive in area, cost and power consumption. Therefore, we must consider a right QoS (Quality-of-Service) metric before employing these algorithms. Routing algorithms can be fault-tolerant algorithms such as backtracking. In case of progressive algorithms, a channel is reserved before a flit is forwarded. Some routing algorithms send packets/flits only in the direction that is nearer to the destination. These routing algorithms are referred as profitable algorithms. A misrouting algorithm may forward a packet/flit away from the destination as well. Based on the number of available routing paths, routing algorithms can be finally classified as complete and partial routing algorithms. Various routing algorithms have been proposed for the NOC. Most researchers suggested static routing algorithms and performed communication analysis based on the static behavior of NOC processes, thus, determining the static routing for NOC [15]. Siebenborn and Hu used a CDG (Communication Dependency Graph) to analyze inter-process communications [16]. Most NOC implementations used either XY routing or street sign routing algorithms. In a comparison of deterministic (dimension-order) and adaptive routing algorithms for mesh, torus, and cube networks was presented [17]. Mello researched the performance of minimal routing protocol in NOC [18]. They concluded that the minimal routing provided better results than adaptive routing for on-chip-communications, as the adaptive routing concentrates on the traffic in the center of the NOC.

V. Switching Techniques

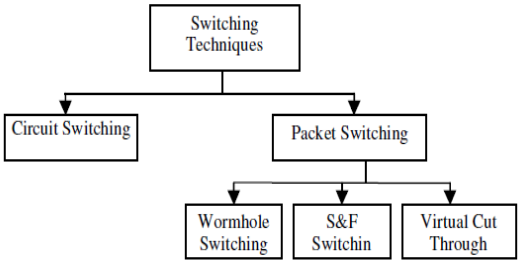

Fig. 2. Switching techniques

Switching techniques can be classified based on network characteristics. Circuit switched networks reserve a physical path before transmitting the data packets, while packet switched networks transmit the packets without reserving the entire path. Packet switched networks can further be classified as Wormhole, Store and Forward (S&F), as shown in Fig.2 and Virtual Cut through Switching (VCT) networks. In Wormhole switching networks, only the header flit experiences latency. Other flits belonging to the same packet simply follow the path taken by the flit. If the header flit is blocked then the entire packet is blocked. It does not require any buffering of the packet. Therefore, the size of the chip drastically reduces. However, the major drawback of this switching technique is a higher latency. Thus, it is not a suitable switching technique for real-time data transfers. Al-Tawil provided a well-structured survey of Wormhole Routing techniques and its comparison with other switching techniques [19]. S&F switching forwards a packet only when

there is enough space available in the receiving buffer to hold the entire packet. Thus, there is no need for dividing a packet into flits. This reduces the overhead, as it does not require circuits such as a flit builder, a flit decoder, a flit stripper and a flit sequencer. Nevertheless, such a switching technique requires a large amount of buffer space at each node. Thus, it may not be a feasible solution for embedded applications. The CLICHÉ implementation of a NOC is an example of store-and-forward switching. Millberg employed this switching technique in their Nostrum NOC implementation [20]. In VCT switching, a packet is forwarded to the next router as soon as there is enough space to hold the packet. However, unlike S&F, the VCT algorithm divides a packet into flits, which may be further divided into piths. Therefore, it has the same buffer requirement as S&F. None of the NOC implementations has adopted this switching technique in their implementation. Ad-hoc switching techniques can be also developed by combining different switching techniques. For instance, VCs can be used for each class of traffic, while each channel is operated as per the principles of circuit switching. The Ethereal and Mongo NOC implementations use such a combination of techniques.

VI. Flow Control

Flow control determines how network resources, such as channel bandwidth, buffer capacity, and control state, are allocated to a packet traversing the network. The flow control may be buffered or buffer less (see Fig. 3). The Buffer less Flow Control has more latency and fewer throughputs than the Buffered Flow Control. The Buffered Flow Control can be further categorized into Credit Based Flow Control, ACK/NACK Flow Control, STALL/GO Flow Control, T-Error Flow Control, and Handshaking Signal based Flow Control.

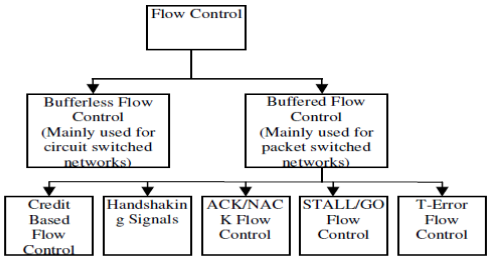

Fig. 3. Flow control techniques

In Credit Based Flow Control, an upstream node keeps count of data transfers, and thus the available free slots are termed as credits. Bjerregaard and Sparso have proposed the design and implementation of a virtual channel router using asynchronous circuit techniques [21],[22]. Once the transmitted data packet is either consumed or further transmitted, a credit is sent back. Bolotin used Credit Based Flow Control in QNOC [23], [24]. In Handshaking Signal Based Flow Control, a VALID signal is sent whenever a sender transmits any flit. The receiver acknowledges by asserting a VALID signal after consuming the data flit. Zeferino used handshaking signals in their SoCIN NOC implementation [25]. In the ACK/NACK protocol a copy of a data flit is kept in a buffer until an ACK signal is received. On assertion of ACK, the flit is deleted from the buffer; instead if a NACK signal is asserted then the flit is scheduled for retransmission. Bertozzi, Benini, and Micheli used this flow control technique in their XPIPES implementation [26]. In the STALL/GO scheme, two wires are used for flow control between each pair of sender (producer) and receiver (consumer). When there is an empty buffer space, a GO signal is activated. Upon the unavailability of buffer space, a STALL signal is activated. None of the present NOC implementations have employed this flow control scheme. The T-Error Flow Control scheme is very complex as compared to other flow control mechanisms. It aims at enhancing the performance at the cost of reliability. Real time systems operating in a noisy environment must avoid the use of this flow control mechanism. None of the present NOC implementations has employed this flow control scheme.

VII. Virtual Channel

The design of a virtual channel (VC) is another important aspect of NOC. A virtual channel splits a single channel into two channels, virtually providing two paths for the packets to be routed. There can be two to eight virtual channels. The use of VCs reduces the network latency at the expense of area, power consumption, and production cost of the NOC implementation. However, there are various other added advantages offered by VCs. Network deadlock/ live loc : Since VCs provide more than one output path per channel there is a lesser probability that the network will suffer from a deadlock; the network live lock probability is eliminated (these deadlock and live lock are different from the architectural deadlock and live lock, which are due to violations in inter-process communications). Performance improvement: A packet/flit waiting to be transmitted from an input/output port of a outer/switch will have to wait if that port of the router/switch is busy. However, VCs can provide another virtual path for the packets to be transmitted through that route, thereby improving the performance of the network. Supporting guaranteed traffic: A VC may be reserved for the higher priority traffic, thereby guaranteeing the low latency for high priority data flits [23], [30]. Reduced wire cost: In today's technology the wire costs are almost the same as that of the gates. It is likely that in the future the cost of wires will dominate. Thus, it is important to use the wires effectively, to reduce the cost of a system. A virtual channel provides an alternative path for data traffic, thus it uses the wires more effectively for data transmission. Therefore, we can reduce the wire width on a system (number of parallel wires for data transmission). For example, we may choose to use 32 bits instead of 64 bits. Therefore, the cost of the wires and the system will be reduced.

VIII. Buffer Implementation

A higher buffer capacity and a larger number of virtual channels in the buffer will reduce network contention, thereby reducing latency. However, buffers are area hungry, and their use needs to be carefully studied and optimized.

Bolotin proposed a simple implementation of buffer architecture for NOC [27]. The Proteo implementation of buffer architecture has been described in [28]. Gupta studied the trade-off between buffer size and channel bandwidth to secure constant latency. They concluded that increasing the channel bandwidth is preferable to reducing the latency in NOC.

IX. Error Correction and Decoding

The need for implementation of fault tolerant, error detection, and error correction techniques is not certain for on chip implementations. Frederico, Santo, and Susin proposed a fault tolerant routing protocol for NOC [14]. Bolotin in their implementation of QNOC [23], [24] argued that the communication strategies for on chip network may be considered reliable, while [29] proposed a fault tolerant routing algorithm and fault tolerant flow control techniques for NOC architecture respectively. Zimmer, Jantasch, and Bertozzi, Binini, and Micheli proposed error detection and correction schemes for data on NOC links .

X. Network Interface

The network interface (NI) is responsible for packetization and depacketization of data traffic, in addition to conventional interfacing. This functionality may be implemented either with hardware or with software. Bhojwani and Mahapatra compared software and hardware implementations of NI [30]. They showed that the software implementation of NI takes about 47 cycles to complete packetization/depacketization, while the hardware version takes only 2 cycles. Substantial research has been conducted to propose the right data formats needed for various layers in the protocol stack. Ethereal and XPIPES NOCs use the OCP protocol, while SPIN and Proteo NOC have integrated the Virtual Component Interface (VCI) protocol in their implementations.

XI. QoS

New algorithms have been proposed in this domain to reduce power consumption and area requirements while securing cost optimization. One of the main concerns in NOC is to be able to reduce the latency of operation. Therefore, there are various levels of latency metrics that may be offered. Router architectures for supporting GT (Guaranteed Bandwidth) and BE (Best Effort) services have been proposed [10].

XII. Arbitration Techniques

A NOC, which is capable of supporting different classes of service levels such as best effort and guaranteed traffic, needs to support an arbitration mechanism. This arbitration mechanism schedules a flit for transmission on the output path. There are various arbitration mechanisms such as RR (Round Robin), FCFS (First Come First Serve), PB (Priority Based), and PRBB (Priority Based Round Robin). Usually, RR and FCFS are used for best effort data flits and PB or PBRR is used for guaranteed traffic. SPIN and RASoC implemented RR arbitration [8].

XIII. Architectural Issues

A NOC system may be categorized based on the customization and parameterization capabilities embedded in its architecture. NOC architecture may further be defined as a homogenous architecture or a heterogeneous architecture. A heterogeneous architecture will have a fixed topology and cannot be customized as per an application requirement. Therefore, the design time with such architecture will be less. However, a homogenous architecture may be customized each time as per the application requirement and may be more efficient in terms of area, power and latency of operation. Most of the NOC implementations support a homogenous architecture while the XPIPES supports a heterogeneous architecture [26], [28]. Most researchers have focused on the communication architecture of NOC. Binini and Micheli mapped the OSI layered architecture onto a NOC for on-chip communication. Agarwal and Shankar focused on exploiting the computing capability and provided a layered architecture for system design. Their layered architecture consists of mapping different domains such as applications, algorithms, RTOS and protocol on a NOC environment. Most of the NOC architectures have yet to be implemented for commercial purposes. The Arteris NOC is an example of a commercially available NOC [3]. Arteris has developed a NOC complier for targeting applications onto its NOC chip. STMicroelectronics has shown interest in NOC implementations as well. Philips has implemented the Ethereal NOC. Once the design of the basic NOC architecture became established, new techniques evolved to address advanced issues such as dynamic load balancing, shortest/fastest data path, and energy-efficient NOC architecture design.

XIV. Conclusion

The Packet-switched networks for communications within large multi core systems on-chip (Network on Chip) concept somehow separates the concerns of computing and communication, and is expected to be ideally suited to address this increased system complexity and declining system productivity. Researchers have well addressed its architectures and hardware-related issues. Still, an integrated approach for modeling, co-designing and co-developing HW-SW with a NOC architecture is continue in research. We need to research low cost, area and power efficient solutions of NOC for it to be applicable in the embedded systems.

References

[1] Jantsch and H. Tenhunen, "Network On Chips," *I s ted. Kluwer Academic Publishers*, Boston, 2003 ch 1.

[2] S. Kumar, A. Jantsch, J-P. Soininen, M. Forsell, M. Millberg , J. Oberg, K. Tiensyrja, and A. Hemani, "A Network On Chip

Architecture and Design Methodology", *IEEE Computer*, pp. 117-124, 2002.
[3] ARTERIS. 2005. "A Comparison of Network-On-Chip and Buses." [Online]. *White paper. http://www.arteris.com/noc whitepaper.pdf.*
[4] K. Emerson, "Asynchronous Design - An Interesting Alternative," in *Proc. 10th International IEEE Conference on VLSI Design*, 1997, pp. 318-320.
[5] J. Dally and B. Towles, "Principles and Practices of Interconnection Networks ," *1 st ed. Morgan Kaufmann*, 2004 ch 1.
[6] P. Pratim Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, "Performance Evaluation and Design Trade-offs for Network-On-Chip Interconnect Architectures," in *IEEE Transactions on Computers*, 2005, vol. 54, no. 8, pp. 1025-1040.
[7] D. Kim, Manho Kim, and G.E. Sobelman, "CDMA-Based NOC Architecture," in *Proc. IEEE Conference on Circuits and Systems*,2004 vol. 1, pp. 137-140.
[8] Adriahantenaina, H. Charlery, A. Greiner, L. Mortiez, and C. A. Zeferino, "SPIN: A Scalable, Packet Switched, On-Chip Micro-Network", in *Proc. IEEE Conference on Design, Automation and Test*, 2003 pp. 70-73,.
[9] F. Karim A. Nguyen, and S. Dey, "An Interconnect Architecture for Networking Systems On Chips", *IEEE Journal on Micro High Performance Interconnect*, Sept 2002, vol. 22, issue 5, pp. 36-45,.
[10] E. Rijpkema, K. Goossens, A. Radulescu, J. Dielissen, J. van Meerbergen, P. Wielage, and E. Waterlander, "Trade-offs in The Design of A Router with Both Guaranteed and Best-Effort Services for Networks On Chip," *IEEE Proc. on Computers and Digital Techniques* , September 2003, vol. 150, Issue 5, pp. 294-302.
[11] Kumar, D. Manjunath, and J. Kuri, *Communication Networking: An Analytical Approach* , 1 st ed. Morgan Kaufmann, 2004, ch 1.
[12] P. T. Wolkotte, G. J. M. Smit, G. K. Rauwerda, and L. T. Smit, "An Energy-Efficient Reconfigurable Circuit Switched Network-On-Chip," in *Proc. 19th IEEE International Conference on Parallel and Distributed Processing Symposium*,2005 , pp. 155-163.
[13] J. W. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," in *Proc. IEEE International Conference on Design and Automation,* , June 2001, pp. 684-689.
[14] C. Albenes, Z. Frederico, G. M. E. Santo, A. A. Susin, and S. Parl , "A Parameterizable Interconnect Switch for Networks-On-Chips," in *Proc. ACM Conference*, 2004, pp. 204-209.
[15] Siebenborn, O. Bringmann, and W. Rosenstiel, "Communication Analysis for Network-On-Chip Design," in *Proc. IEEE International conference on Parallel Computing in Electrical Engineering,2004*, pp. 315-320.
[16] J. Hu and R. Marculescu, "Energy-Aware Communication and Task Scheduling for Network-On-Chip Architectures Under Real-Time Constraints," in *Proc. IEEE Conference Design Automation and Test in Europe,* 2004, vol. 1, pp. 234-239.
[17] C. Neeb, M. Thul, and N. Andwehn "Network On-Chip-Centric Approach to Interleaving in High Throughput Channel Decoders," in *Proc. IEEE International Symposium on Circuits and Systems,* 2005 pp. 1766–1769.
[18] F. Moraes and N. Calazan, (Oct 2004) "An Infrastructure for Low Area Overhead Packet-Switching Network On Chip," [Online] . *Integration - The VLSI Journal*, vol. 38, Issue 1, pp. 69-93.
[19] K. M. Al-Tawil, M. Abd-El-Barr, and F. Ashraf, "A Survey and Comparison of Wormhole Routing Techniques in Mesh Networks," *IEEE Network*,[Online] vol. 11, 1997, pp. 38–45.
[20] M. Millberg, E. Nilsson, R. Thid, S. Kumar, and A. Jantsch. "The Nostrum backbone - A Communication Protocol Stack for Networks On Chip," in *Proc. IEEE International Conference on VLSI Design*, 2004, pp. 693.
[21] [Online]. *www.scientificjournals.org*
[22] T. Bjerregaard and J. Sparsø, "A Router Architecture for Connection-Oriented Service Guarantees in The MANGO Clockless Network-On-Chip," in *Proc. of IEEE on Design Automation and Test*, 2005 vol. 2, pp. 1226-1231.
[23] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "QNoC: QoS Architecture and Design Process for Network On Chip," *Journal of Systems Architecture*,[Online] Volume 50, Issue 2-3 (Special Issue on Network on Chip), February 2004, pp. 105-128.
[24] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "Cost Considerations in Network On Chip," Integration*: The VLSI Journal*, [Online] no. 38, 2004, pp. 19-42.
[25] C. A. Zeferino and A. A. Susin, "SoCIN: A Parametric and Scalable Network-On-Chip," in *Proc. 16th Symposium on Integrated Circuits and Systems Design,* , 2003, pp. 169-175.
[26] D. Bertozzi and L. Benini, "Xpipes: A Network-On-Chip Architecture for Gigascale Systems-On-Chip," *IEEE Circuits and Systems Magazine*,2004, vol. 4, Issue 2, pp. 18-31.
[27] E. Bolotin, A. Morgenshtein, I. Cidon, R. Ginosar, and A. Kolodny, "Automatic Hardware-Efficient SoC Integration by QoS Network-On-Chip," in *Proc. 11th International IEEE Conference on Electronics, Circuits and Systems*, 2004, pp. 479-482.
[28] Saastamoinen, M. Alho, and J. Nurmi, "Buffer Implementation for Proteo Network-On-Chip," in *International IEEE Proceeding on Circuits and Systems*, May 2003, vol. 2, pp. 113-116.
[29] M. Pirretti, G. M. Link, R. R. Brooks, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin, "Fault Tolerant Algorithms for Network-On-Chip Interconnect," in *Proc. IEEE Proceeding on Computer Society* February 2004, pp. 46-51.
[30] P. Bhojwani and R. Mahapatra, "Interfacing Cores with On-Chip Packet-Switched Networks," in *Proc. 16th International IEEE Conference on VLSI Design*, 2003, pp. 382–387.

R. Mutha born in Jodhpur, Rajasthan in august 1978. She had received M.Tech. in computer science and engineering Degree from Bhagwant University, Ajmer., Rajasthan , India. And Master degree in computer science from Guru Jambheshwar University, Hissar, Haryana,India.

She has worked as Assistant Lecturer at Jodhpour Institute of Engineering and Technology. She has published 2 research papers in International Journals and have presented 8 papers in International and National Refered Journals and Confernces. Her area of interest is Ad-Hoc Networking. She is currently working on networking research area for future research and development and new initiations in this fields.