



# **Packet Switching – Queuing Architecture: A Study**

Shikhar Bahl<sup>1</sup>, Rishabh Rai<sup>2</sup>, Peeyush Chandra<sup>3</sup>, Akash Garg<sup>4</sup>

M.Tech, Department of ECE, Ajay Kumar Garg Engineering College, Ghaziabad, U.P., India<sup>1,2,3</sup>

M.Tech, Department of CSE, Ajay Kumar Garg Engineering College, Ghaziabad, U.P., India<sup>4</sup>

**ABSTRACT:** In this paper, the packet switching architecture with output queuing is used. Here the switch is internally non-blocking, but if packets destined to same outputs, output blocking will occur and (even if there are output queues) has a capacity of  $N \times N^2$ . An exact model of the switch has been developed which can be used to determine the blocking performance of the switch and obtain both its throughput and packet loss characteristics. In this architecture, each line card is connected by a dedicated point to point link to the central switch fabric. Two structures can be classified as Centralized and Distributed. Buffer arrangements are also categorized into output queued and combined input- output queued switches and hardware complexity of OQ, VOQ are also discussed.

**KEYWORDS:** IQ Switches, OQ Switches, VOQ Switches, Switch Fabric.

## **I. INTRODUCTION**

Over the past decade, data communication has been revolutionized by a radically new technology called Packet Switching. Packet switched routers need buffers during times of congestion. Buffer arrangements can be placed on inputs and outputs within the switch fabric. A major challenge connected to high-speed switching is related today to switch design that requires the best possible ease of implementation and good performance. The main functions of Switching are: (i) manage packet buffering while selecting packets to transmit each time to avoid contentions and cell loss. (ii) Routing packets from their incoming ports to their destination ports. To avoid contentions and cell loss, the incoming packets are stored in buffers. These buffers can be in inputs, in outputs, in inputs and outputs or shared by inputs and outputs. Although, a lot of existing switches use the shared buffers techniques. In OQ (Output Queue) strategy, all incoming cells at the input are allowed to arrive at the output port and they are served using FIFO (First in First Out) discipline. In this strategy, there is 100% throughput and there is an internal speedup of order of  $N$  (impractical for large  $N$ ) [1]. An output buffered switch can be more complex than an input buffered switch because the switch fabric must operate at much higher speed to reduce the probability of cell loss.

The IQ (Input Queue) switch can transfer at most one packet from an input and at most one packet to an output in a slot. At the beginning of a slot, more than one HOL (Head of Line Blocking) cells from the input queues have the same destination, and then only one of them is switched and transmitted on the output link in the slot. These switches are easy to implement and throughput is limited to 58.6% under uniform traffic. The other HOL cells continue to be queued at their inputs. Thus, packets in an IQ switch can experience HOL blocking, in which a blocked HOL cell blocks the cells behind it in the input FIFO queue even though the destination ports of these other cells are free and are idling. One of the proposed solution for IQ switches which is VOQ. In this, the main aim is to overcome HOL blocking as there is no speed up requirement. It needs scheduling algorithms to resolve contention problems (i) complexity (ii) performance guarantee. Different scheduling algorithms for VOQ switches are considered most of them achieve 100% throughput under uniform traffic, but the throughput is reduced under non-uniform traffic.

## **II. LITERATURE SURVEY**

J. Xian and Ch.-T. Lea et.al, 1991 [1] proposed a model in which they operated a technique called parallel iterative matching, which can quickly identify a set of conflict free cells for transmission in a time slot and a more flexible approach to high availability using multiple redundant paths between hosts. T. Anderson et.al, 1994 [2] used an  $N \times N$  port input- queued switch with FIFO and according to him, the throughput is limited. J. H.J. Chao, C.H. Lam and E.

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

Oki et.al, 1998 [3] proposed a model in which they found the optimal division between the input and output queues under different (space, time) combination and how that division shifted with the (space, time) implementation. N. McKeon, A. Mekittikul, V. Anantharam and J. Walrand et.al, 1999 [4] proposed that bandwidth of optical fibers caused tremendous increase in speed of data transmission and a bit sliced crossbar fabric to switch packets at 10 Gb/s at inputs and outputs. H.J. Chao et.al, 2001 [5] proposed a model by designing a high capacity packet switch (eg. Multiple terabits/second) that: (1) supports individual line rates in excess of the speeds of available electronic memory and is capable of supporting the same qualities and service as an output-queued switch and from a more practical point of view, the arbitration should be separated from the output packet scheduling to keep the implementation and time complexities reasonable. Using speedup with input-output queuing is widely accepted as the most feasible solution to building large-scale switches and they also supported idea to emulate an input-output buffered switch with speedup as a purely output queued switch by specially scheduling cells from inputs to outputs such that each output is identical to the emulated purely output queued switch.

### III.GENERAL ARCHITECTURE

Here the OQswitch, the fabric provides a dedicated point to point channel between each input and output. The figure-1 shows the switch architecture with output queuing is being used.

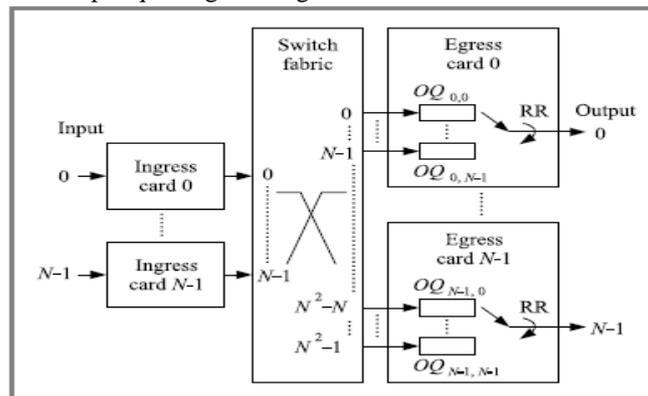


Figure-1. The Switch Architecture with Output Queuing.

In this figure, there are N input ports, N output ports and the switch fabric. They are implemented on separate ingress and egress card 0 to N-1. An input to ingress card is connected to switch fabric by one line, and this enables the switch to simultaneously transfer up to N packets to each output port. The figure-1 shows, at most N packets arrive during a time step, all packets are transferred to their respective outputs in the switching phase immediately following their arrival. At the output port side (OQ<sub>i,i</sub>), each packet received from the input side stored in the output buffer receiving N packets from input port through the switch fabric. Therefore these packets can be simultaneously written to N-1 queues.

### IV.SWITCHING FABRIC ARCHITECTURE

Now, we will consider the switching fabric architecture. The switch fabric has a capacity of  $N \times N^2$  and it uses a non-blocking switch fabric. There are configurations for using the switch fabric. It can reside on the backplane and can be connected directly with line cards or it can reside on a separate card attached to it connected through a mesh to all line cards. In central case, the switch fabric constitutes one module produced on several boards. They are simple on input and output and are arranged on line cards. The centralized approach requires each input to contact a centralized scheduler at every arbitration cycle. With N ports, N request must be connected to line cards and processed by the arbiter each cycle.

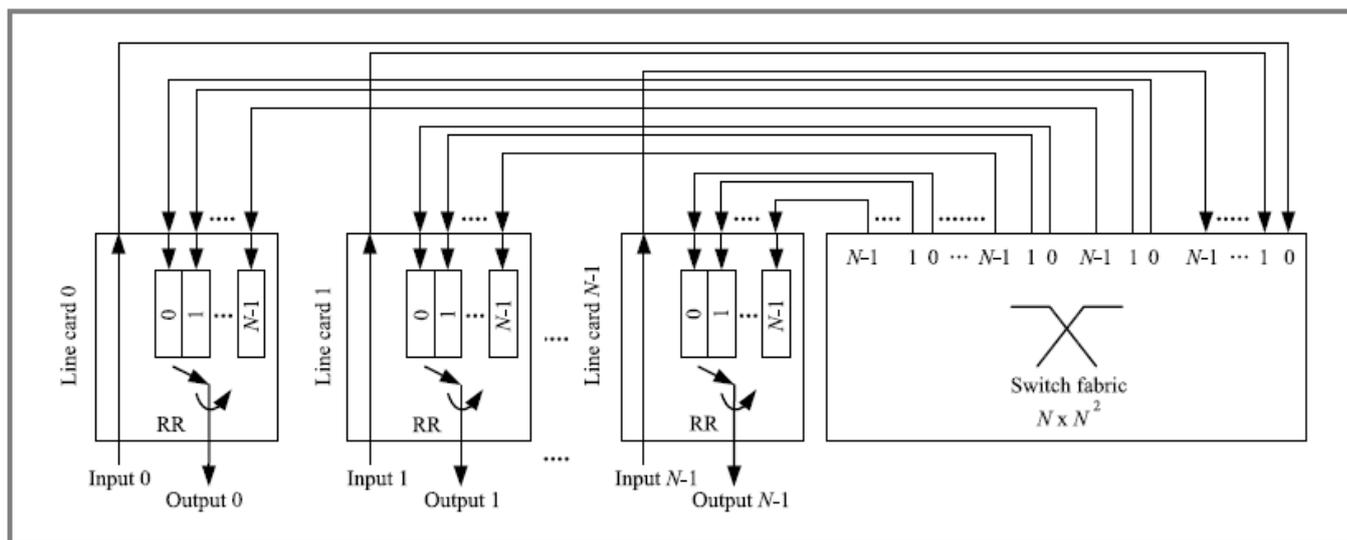


Figure-2. The Switch Architecture with Centralized Switch Fabric

This requires a high speed control path running at the line rate between every input and the central scheduler. In this case, the line cards are connected with the switch fabric by 2 lines, but the switch fabric will require more boards with buffer memories. The switch fabric of size  $N \times N^2$  can also be decentralized by putting each  $1 \times N$  segment on each line card.

In centralized mode, the switch fabric is located on several ingress/ egress cards. The figure-2 shows a switch fabric on switch architecture of  $N \times N^2$  switch fabric, so that each row of crossbar switch fabric is placed on line cards [2]. The goal is to work on architecture that is scalable, both in terms of line speed as well as in terms of number of ports, at the same time flexible and reliable and can support different line cards. This requires a number of outputs from linecards. Each incoming and outgoing lines are connected to line cards and connections made between them act as buses and are broadcasted from inputs to all outputs. The function of address filter (AF filter) is in the given connection whether the packets in line cards are destined to output or not. The number of Address Filters is  $N \times N$  size and it can operate with line speed.

When the traffic is uniformly distributed, servicing the maximum number of queues leads to 100% throughput. When it is non-uniform, some queues become longer than others. A good algorithm keeps the queue lengths matched, and service a large number of queues. The maximal weight algorithm in input queuing depends on the Fair Access Round Robin (FARR) and Longest Port First (LPF). For very low bandwidth systems, the choice of a shared Bus or shared-memory architecture is obvious. The result is a fully synchronous serial backplane that needs no phase reacquisition, after a switch reconfigures and therefore maximizes the usable bandwidth. Furthermore, some synchronous crossbar fabrics also have the arbitration and configuration control logic integrated in the switch.

## V. BUFFER IMPLEMENTATIONS

The buffering operations consist in queuing the cells to transmit. The performances of the switch can be affected differently according to the way that is done. Different strategies are used depending on the physical site of the queues in both inputs and outputs, or are shared buffers. In this architecture, every output can receive simultaneously during the same cycle,  $N$  cells from the  $N$  inputs. Thus, the switch must be able to put in the same queue and during the same cycle,  $N$  cells from the  $N$  inputs. Packets from  $N$  queues in each output Port are readout using Round-Robin algorithm pointer denoted by RR, and modified in such a way, when all packets from the same position are already readout, the RR is set back to zero. In given figure-3, 3 time slots 1,2 and 3 from the switching fabric are shown. Packet Switching architecture uses a crossbar for interconnecting the cards on which ports reside. Despite the  $N^2$  complexity, a cross-bar is actually the most popular switch core.

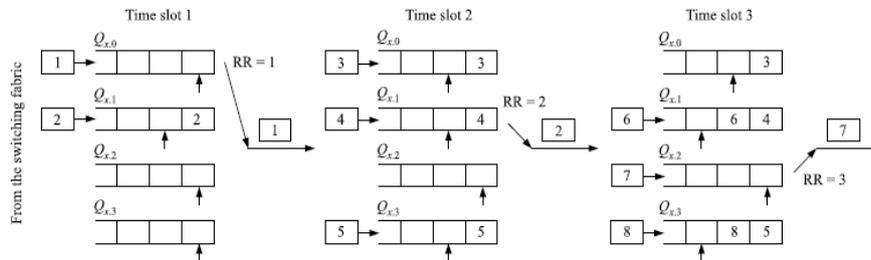


Figure-3. The Example of Buffer Operation with Separate Pointers.

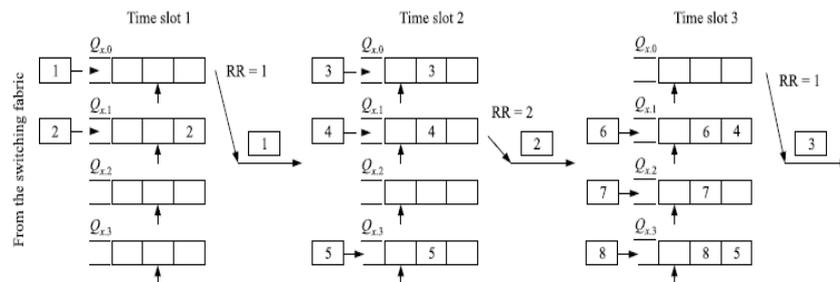


Figure-4. The Example of Buffer Operation with One Pointer

The figure-3 shows the buffer operation with separate pointers shown by arrow signifies the state of the pointers in different time slots 1, 2 and 3. In time slot 1, the packets from the switching fabric 1 and 2 from inputs  $Q_{x,0}$  and  $Q_{x,1}$  is directed to output and  $RR$  is set to 1 i.e. ( $RR=1$ ) and packet 2 from input 1 is stored in  $Q_{x,1}$  [3]. In the next time slot 2, the packets from the switching fabric numbered as 3, 4 and 5 from inputs 0, 1 and 3 are directed to output and  $RR$  is set to 2 and packet 2 from  $Q_{x,1}$  is sent out. In next time slot 3, the packets from the switching fabric numbered as 6, 7 and 8 from inputs numbered as 1, 2 and 3 is immediately considered to output  $x$  and packet 7 is sent to the output. Packets 6 and 8 are stored in  $Q_{x,1}$  and  $Q_{x,3}$ . In the other case, the figure-4 shows buffer operation with one pointer is assigned to all queues.

In time slot 1, the packets from the switching fabric 1 and 2 from inputs 0 and 1 are immediately sent to output  $x$  and Round Robin ( $RR$ ) is set to zero. (Since  $HOL$  packet from  $Q_{x,0}$  has highest priority) as shown in figure-4, the packet 1 from input 0 is sent to output and packet 2 is stored in  $Q_{x,1}$ . In next time slot 2, the packets from switching fabric 3, 4 and 5 from inputs 0, 1 and 3 and the packet 2 from  $Q_{x,1}$  is sent out. After this packet is sent out there is not other any packet in first memory cell. During the next time slot 3, packets from switching fabric 6, 7 and 8 from inputs numbered as 1, 2 and 3 and since  $RR$  is set to zero and packet 3 from  $Q_{x,0}$  is sent immediately to the output. In next 3 time slots, the packets from switching fabric numbered as 4, 5 and 6 will be sent out to the output.

For the packet switching there is no special transport path established for a connection and variable length data packets carry information used by network nodes in making forwarding decisions. There is no signaling needed for connection setup. Forwarding tables in the network nodes are updated by routing protocols. The best effort service for all connections in conventional packet switched networks. The main problem of IQ (Input Queue) switching is  $HOL$  blocking, which can have a severe effect on throughput. It is well known that if each input maintains a single FIFO, then  $HOL$  blocking can limit the throughput to 58.6% [4]. One method is that has been proposed to reduce  $HOL$  blocking is to increase the speedup of a switch. A switch with a speed up of  $S$  (say) can remove upto  $S$  packets from each input and deliver upto  $S$  packets to each output within a time slot, where a time slot is the time between packet arrivals at input ports. Hence, an OQ switch has a speed up of  $N$  while an IQ switch has a speed up of 1. For values of  $S$  between 1 and  $N$  packets need to be buffered at the inputs before switching as well as at the outputs after switching.



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

We call this architecture a combined input and output queued (CIOQ) switch. In practice, we are not only interested in the throughput of a switch, but also in the latency of individual packets. Packets in an IQ switch not only contend for an output, they also contend for an entry into the switch fabric with packets that are destined for other outputs. This phenomenon is input contention. Each input can deliver only one packet into the fabric at a time; if it has packets for several free outputs, it must choose just one packet to deliver, holding other packets back. This place, a packet at the mercy of other packets destined for other outputs. This is in a stark contrast with output-queuing, where a packet is unaffected by packets destined for other outputs. CIOQ switches make no guarantees about the delay of an individual packet; instead they consider only average delay and throughput. While these switches are academically interesting, they give us the principle benefit of output queuing: the ability to control the delay of individual packets [5]. Rather than final values of speedup that work well on average, or with simplistic and unrealistic traffic models, we find the minimum speedup such that a CIOQ switch behaves identically to an OQ switch.

Although output buffers give the optimal delay-throughput performance, switches that use them are difficult to achieve. In output buffer method, every output can receive simultaneously during the same cycle, N cells from the N inputs. Thus, the switch must be able to put in the same queue and during one cycle, the N cells destined to the same output. The operation of setting in queue must therefore operate N times quickly than the rate of all arrivals (speed up). If, this solution is feasible in case of small capacity switches, it should not be possible for switches of big capacities (the N-times speed-up in the switch limits the scalability of this architecture). A major drawback of input buffer is related to queue managing while selecting cells to transmit at every cycle. The simplest way consists in storing the incoming cells in FIFO queues.

## V.COMPARISION TABLES

The differences between circuit switching, datagram packet switching and virtual circuit packet switching are shown in table 1 in which there is a dedicated path in circuit switching, there is no dedicated path in datagram packet switching and in virtual circuit packet switching, also there is no dedicated path. The differences between input queueing, output queueing and combined input and output queueing are shown in table 2 in which packets are stored at the input side in input queueing, packets are stored at the output side in output queueing and in combined input and output queueing packets are stored at both input and output sides. In input queueing switch speed is equal to line speed, in output queueing switch speed is equal to N times of line speed and in combined input and output queueing there is speedup between 1 and N. There is no overlapping in case of input queueing, in output queueing there is a redundant constraint and in combined input and output queueing there is shared memory switch. The hardware complexities of different buffer strategies are shown in table 3. The parameters like switch fabric speed in output queueing is N, in virtual output queueing it is 1, and in multiple output queueing is also 1 which is same as line speed. However, the switch fabric capacity in output queueing is N\*N, in virtual output queueing is N\*N and in MOQ architecture the capacity is N\*N<sup>2</sup>. Among the different buffer strategies the MOQ architecture is effective considering in high speed and high capacity switches. The memory speed in OQ architecture is N, in VOQ architecture, it is 1 and in MOQ architecture, it is also 1.

**Table 1: - Difference between circuit switching, datagram packet and virtual circuit packet switching.**

CIRCUIT SWITCHING	DATAGRAM PACKET SWITCHING	VIRTUAL CIRCUIT PACKET SWITCHING
Dedicated path	No dedicated path	No dedicated path
Call set up delay	Packet transmission delay	Call set up delay, packet transmission delay
No speed or code conversion	Speed or code conversion	Speed or code conversion
Fixed bandwidth	Dynamic bandwidth	Dynamic bandwidth
No overhead bit after call set up	Overhead bits in each packet	Overhead bits in each packet
Path established for entire connection	Route established for each packet	Route established for entire connection

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

**Table 2:- Difference between input queuing, output queuing and combined input and output queuing.**

INPUT QUEUEING	OUTPUT QUEUEING	COMBINED INPUT AND OUTPUT QUEUEING
Packets stored at the input	Packets stored at the output	Packets stored at both input and output
Switch speed is equal to line speed	Switch speed is equal to N* line speed	Speedup between 1 and N
Combination of packets can be transferred across a switch	Any N packets intended for an output can be transferred	Packets stored in the switch fabric, every output line reads when they transmit.
Input queued switch has additional rate constraints at the output	For output line stability these constraints will arise in output queued switches	For this queuing these constraints will arise at both input and output
Input queuing do not overlap	Redundant constraint	Shared memory switch
It operates on FIFO	There is 100% throughput	Values of speed up need to be buffered
HOL cells continue to be queued at their inputs	Can be more complex because it operates at a much higher speed	In this queuing there is minimum speed up

**Table 3:- The hardware complexity of different buffer strategies.**

Parameters	OQ	VOQ	MOQ
Switch fabric speed	N	1	1
Switch fabric hardware	N <sup>2</sup>	N <sup>2</sup>	N <sup>2</sup>
Number of buffers	N	N <sup>2</sup>	N <sup>2</sup>
Memory speed	N	1	1
Number of schedulers	-	2N	N
Switch fabric capacity	N*N	N*N	N*N <sup>2</sup>

The similar complexity is needed since each VOQ has to be connected with the scheduler to send request signal when it has a HOL packet. The OQ switches consist of N buffers and have to be N times faster than VOQ switches. Different switches of OQ switches are connected to input and output switches to their ingress and egress cards.

## VI. CONCLUSION

In this paper, the packet switching architecture with output queuing has been used. The hardware complexity of OQ, compared to VOQ switches are also discussed and the considerations of OQ switches in different input-output queuing using buffer implementations are used in either separate chip or in the switch fabric architecture. The difference between different types of switching and difference between input queuing, output queuing and combined input and output queuing are also discussed.

## REFERENCES

1. J. Xian and Ch.-T. Lea, 1991, "Speedup and buffer division in input/output queuing ATM switches", *IEEE Trans. Commun.*, vol. 51, no.7, pp. 1195-1203.
2. T. Anderson et al., 1994, "High-speed switch scheduling for local-area networks", *ACM Trans. Comput. Syst.* vol.11, no.4, pp.319-352.
3. H. J. Chao, C.H. Lam, and E. Oki, 1998, "Broadband Packet Switching Technologies: A Practical guide to ATM Switches in IP Routers", *IEEE Trans. Commun.*, vol.48, no. 5, pp. 326-332.
4. N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, 1999, "Achieving 100% throughput in input-queued switches", *IEEE Trans. Commun.*, vol.47, no.8, pp. 1260-1267.
5. H.J Chao, 2001, "Saturn: a terabit packet switch using dual round-robin", *IEEE Commun. Mag.*, vol. 38, no. 12, pp.78-84.

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

## BIOGRAPHY



**Shikhar Bahl** is pursuing M.Tech from Ajay Kumar Garg Engineering College, Ghaziabad (affiliated to Uttar Pradesh Technical University, Lucknow) in electronics and communication engineering. His areas of interest are Control Systems, Electronic Devices and Circuits and Wireless Communication. He completed his B.Tech from Meerut Institute of Engineering and Technology, Meerut (affiliated to Uttar Pradesh Technical University, Lucknow) in Electronics and Telecommunication Engineering with first class in the year 2013. During his career he had done his industrial training at BSNL and RRVNL. He had done his major work on project in filter design and its specifications in the respective branch for consecutive four years during his course.



**Rishabh Rai** is pursuing M.Tech from Ajay Kumar Garg Engineering College, Ghaziabad (affiliated to Uttar Pradesh Technical University, Lucknow) in VLSI Design. His areas of interest are Digital System Design, Embedded System Design & Low Power VLSI Design. He completed his B.Tech from Vishveshwarya Institute of Engineering & Technology, Gr. Noida (affiliated to Uttar Pradesh Technical University, Lucknow) in Electronics and Telecommunication Engineering with Honours in the year 2013. During his career, he has been appreciated & certified in 2013, for his academic performance & excellence in the respective branch for consecutive four years during B.Tech and received Amul Vidya Bhushan Award in 2009, for his academic performance & excellence in AISSCE-2009, at the District Level. During his graduation, he has also been awarded with the Best Paper Presentation in the IEEE sponsored National Conference; ETEAT-2013. He has Published 07 Technical/Research Papers/Abstract in several National/International Conferences/Journals till date. He is the student member of IEEE (Membership Number: - 92416730).



**Peeyush Chandra** is pursuing M.Tech from Ajay Kumar Garg Engineering College, Ghaziabad (affiliated to Uttar Pradesh Technical University, Lucknow) in electronics and communication engineering. His areas of interest are Digital Electronics, Electronic Devices and Circuits and Wireless Communication. He completed his B.Tech from Institute of Technology & Management, Gorakhpur (affiliated to Uttar Pradesh Technical University, Lucknow) in Electronics and Telecommunication Engineering with first class in the year 2012. During his career he had done his industrial training at BSNL. He completed his major work on project in Infrared Remote Control based Home Appliances with Fan speed control.



**Akash Garg** is pursuing M.Tech from Ajay Kumar Garg Engineering College, Ghaziabad (affiliated to Uttar Pradesh Technical University, Lucknow) in Computer Science & Engineering. His areas of interest are Java, C, C++, Intrusion Detection System, Neural Networks, and Data Mining. He completed his B.Tech from Mangalayatan University, Aligarh in CSE with First Class in the year 2012. During his career he had done his industrial training at Technobyte. He completed his major work on project in JSP based FriendzPlanet. He is the student member of IEEE (Membership Number: - 93283727).