# Permuted Diagonal Maximum Weight Matching (PDMWM) Scheme for Cell Scheduling in Fixed Length Packet Switches

**Sunil Kore[1], Ajinkya Biradar[2], Sayali Kore[3]**

**Electronics Department, Walchand College of Engineering, Sangli,**
**Maharashtra 416415, India**
***[1]snkore@yahoo.com***
***[2] biradarajinkya@gmail.com***
***[3] contactsayalikore@gmail.com***

**Abstract**

Explosive growth in internet is demanding very fast switching fabric in internet routers and switches. Packets need to be buffered at input or output or on both sides of crossbar switching fabric. Crossbar switches are used for switching because of no bandwidth limitation & high scalability. Its well-known fact that buffering of packets on outside of switch demands switching fabric to be ‘N’ time faster whereas buffering packets on input side limits throughput to 58%. Combined input output queued (CIOQ) switch demands that switch fabric to run at speed up of 2. Hence VOQ (Virtual output Queue) i.e. ‘N’ queues per input port i.e. total $N^2$ queues on input side are suggested to resolve problem of throughput limitation of 58% in input queued switches. In VOQ throughput achieved is 100%. Selection of packets is key issue in VOQ various schemes like, MWM, MSM and maximal matching, are suggested by researchers in last two decades to improve the performance in terms of throughput and delay. We are addressing Permuted Diagonal Maximum Weight Matching (PDMWM) Scheme which provides 100% instantaneous throughput in each slot under heavy traffic conditions & improves delay performance. Our scheme PDMWM is computationally complex for large size switches but it outperforms at lower size switches and provides optimal performance nearer to output queued switch.

***Keywords:*** *fixed length packet switches, Laplace/Leibniz formula, LQPS, PDMWM, VOQ, WMAX*

## 1. Introduction

Internet services are growing faster and faster and it’s becoming part of everybody’s life. Demand for fast internet service is increasing pressure on router design engineers to provide faster switching architectures. Basically cross bar switches are used in internet router/switches because of scalability and no bandwidth limitation exist with it. Present switches use crossbar switches along with buffering of packets either on input-output or on combined input output. Virtual output queues inn which packets are buffered or input side destined for each output separately [2], [3], and [5]. This demand for $N^2$ queues on input side. VOQ technique has resolved problem of throughput limitation of 58% where single input queue is used [1], [2], and [7].

VOQ suffers from scheduling of packets. There are total $N^2$ packets at HOL and we need to choose N non-conflicting packets delivered to the output to achieve throughput of 100% in each time slot.

Various packet scheduling schemes are suggested and generally classified as Maximum Size Matching, Maximum Weight Matching, Maximal Matching and Maximal Matching with iteration and Maximal Weight matching with and without iteration [1], [5].

Maximum size matching guarantees for instantaneous throughput to be 100% but do not guarantee for good delay performance. Maximum weight matching scheme gives the best performance equivalent to output queued switch but they are very complex in implementation.

Maximal matching provides good throughput performance and poor delay performance. Iterative maximal matching and its variant such as i-slip; DRR, FIRM, SRA, etc. provide good throughput and delay performance in multiple iteration. Expected number of iterations are Olog(N). There are variants such as weighted i-slip such as i-LQF, i-OCF, etc [1], [6], [7]. which gives good delay performance better than simple i-slip but these are also complex to implement in hardware. A maximum matching is largest size matching that can be made on graph whereas maximal matching is matching to which no further edges can be added without first removing an already matched edge. Hence maximum match can be maximal but vice versa is not true.

Lot of research has been carried out to reduce communication overheads reduction in complexity of hardware, stability, scalability, fairness, etc. Still efforts are made by researchers to provide best optimal solution for scheduling policy in selection of packets in VOQ [1], [7], [8]. Our efforts are also to provide scheduling scheme named as Permuted Diagonal Maximum Weight Matching scheme to provide better delay and throughput.

**VOQ Switch:**

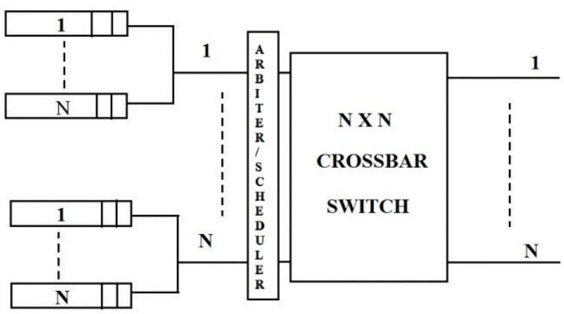



Fig. 1 VOQ Switch

Switch model is as shown in Fig. 1 is NxN switch with multiple input queues used to store the arriving packets to input port 'i' and destined to output port 'j' in $j^{th}$ queue at input port i. Hence there are 'N' queues per input port. There are total $N^2$ queues at input side of switch.

Time assumed to be slotted with each slot equal to transmission time of a cell or packet. In each cell slot we select at most 'N' packets from $N^2$ HOL packets. We have put constraint that at most one HOL packet will be chosen from each input port and at most one packet will be delivered to output port. Hence we constrain pattern I of NxN matrix such that

$$\sum_{i=1}^{N} I_{ij} = \sum_{j=1}^{N} I_{ij} = 1$$

Where, $I_{ij}$ is permutation of indicator matrix. Indicator queue-length matrix K is formed such that $K_{ij}$=1 if Queue-length matrix $L_{ij}$>0 else $K_{ij}$=0. Following figure will demonstrate formation of indicator Queue-length matrix K and permutation of $I_{ij}$ from queue occupancy matrix L.

$$L = \begin{bmatrix} 8 & 15 & 12 \\ 7 & 18 & 6 \\ 0 & 7 & 5 \end{bmatrix} \quad K = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Queue occupancy Matrix - L | Indicator queue-length matrix - K | One permutation of indicator matrix - I

Fig. 2 Demonstration of formation of matrix K and I

Different permutations yield with multiple matching solutions for each of matching policy.

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad I = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Maximum Size Match | Maximal Match

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad I = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Maximum weight, Weight = 31 | Maximal Weight, Weight = 30

No unique solution is obtained. If random choice is made then always we may get suboptimal solution.

Such permutation of indicator matrix used for selection of packets from 'N' HOL cell are 'N!'. Out of such 'N!' combinations one best suited combination should be selected, which provides instantaneous throughput to be 100% i.e. 'N' cells are to be selected per time slot. It's also important to reduce backlogs, which demand to select such 'N' packets from Queues whose Queue-lengths are increasing. Hence scheduling in Virtual output queuing is great computationally complex problem. The performance of switching architecture is determined by performance of arbitration algorithm. General classification of different algorithms evolved by different researchers is summarized below -

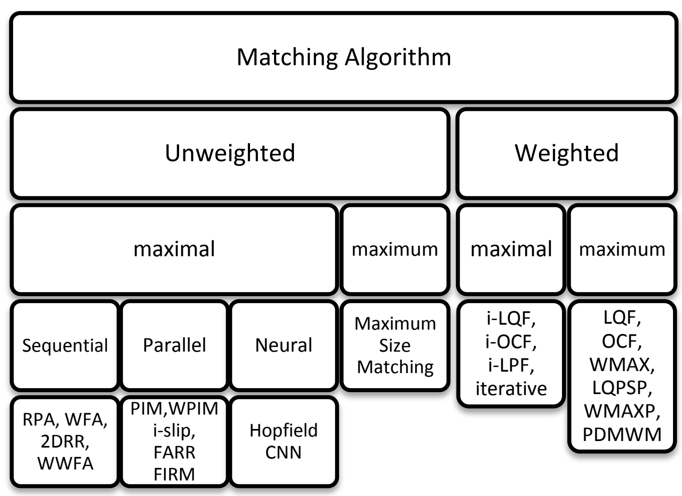



Fig. 3 Classification of scheduling algorithms

## 2. Problem Definition

A scheduling problem in crossbar based VOQ switches can be resolved by finding optimal weight matrix using VOQ occupancies reported in Queue-length matrix L.

$L_{ij}(t)$ = Number of packets backlogged at time(t-1) + arrival if any, at time 't' – departure if any of HOL packet at end of (t-1) timeslot; where $1 \le i, j \le N$.

It is basically constructing bipertite graph G = (V,E) that consist of set V of 2N vertices partitioned into two sets namely 'N' inputs and 'N' outputs. The set of edges E has one edge connecting vertex i of input to vertex j of output for each $L_{ij}(t) > 0$. A matching M on G is any subset of E such that no two edges in M have common vertex. Matching guarantees that only one packet per input and output needs to be transformed [1], [6], [7]. A scheduling policy should work under constrain mentioned above with aiming of instantaneous throughput of 100% and set $M \subset E$ has maximum weight. Hence M must satisfy:

1. Number of edges matched should be 'N'. If no such set exist then select 'M' such that it has maximal matched edges. This condition pulls throughput towards maximum.

2. A match 'M' obtained should have instantaneous average Queue-length > overall average Queue-length and must have variance minimum which is calculated w.r.t. overall average Queue-length $\overline{L}$.

These conditions need to select appropriate permuted diagonals where queues are blowing with higher rate and should be brought under control.

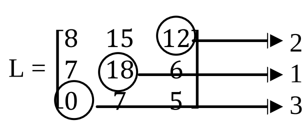
## 3. Cell selection policies

Here we are taking some random queue occupancy as an example to show that how various policies failed to obtain optimal and unique solution.

$$\text{Let } L = \begin{bmatrix} 8 & 15 & 12 \\ 7 & 18 & 6 \\ 0 & 7 & 5 \end{bmatrix}$$

**Random selection:** Select randomly any input and switch the cell from it. Then do not allow this input port and destined port for which cell is switched in current or previous round. It may happen that selected input port has no HOL cell or is having queue occupancy very low. In such cases instantaneous throughput will be reduced backlogs will increase and hamper delay performance. Hence it may or may not be optimal solution [1], [9].

**Longest Queue Priority Selection (LQPS):** It is a greedy policy where highest backlog input is identified and selected first. Again restriction of not selecting the same input & output in remaining round remains the same [1], [9]. This is shown in following selection procedure. Which indicate that numbers of cells switched are less than 'N'




**Longest Queue Priority Selection with Pattern Matching (LQPSP):** In this policy the permuted diagonal is selected and sum of queue occupancy is considered as weight vector [9].

$$W[m] = \sum_{ij} I_{ij}^{m} \odot L_{ij} \tag{1}$$

where m = 1,2,. . ., N!

$I_{ij}^{m}$ is m^th^ permutation of indicator matrix

$\odot$ indicates the point to point multiplication of permuted identity matrix with L matrix.

In case of 3x3 matrix there will be '3!' i.e. 6 elements in weight matrix W[m]. An m^th^ permutation matrix $I_{ij}^{m}$ is chosen to select cell from input HOL Queues.

$$I_{ij}^{1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad I_{ij}^{2} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad I_{ij}^{3} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Mirror

$$I_{ij}^{4} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad I_{ij}^{5} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad I_{ij}^{6} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$L = \begin{bmatrix} \textcircled{8} & 15 & 12 \\ 7 & \textcircled{18} & 6 \\ 0 & 7 & \textcircled{5} \end{bmatrix} \qquad W[m] = [\mathbf{31}, 21, 26, 30, 21, 27]$$

**Weight Maximum with Pattern Matching (WMAXP):** In this policy instead of considering queue-length matrix L a indicator queue-length K matrix is considered, in which

$$K_{ij} = 1 \quad \text{if} \quad L_{ij} > 0$$
$$= 0 \quad \text{otherwise} \tag{2}$$

A weight matrix

$$W[m] = \sum_{ij} I_{ij}^{m} \odot K_{ij} \tag{3}$$

where m = 1,2,. . ., N! For matrix K refer fig.2.
W[m] = [**3**, 2, **3**, 2, **3**, **3**]
Multiple optimal solutions to provide 100% instantaneous throughput are available and randomly one can be chosen. Though instantaneous throughput is 100% but there is no guarantee for good delay performance. Implementation in hardware is easy for smaller size of N [9].

**Permuted Diagonal Maximum Weight Matching Policy (PDMWM):** In given queue occupancy matrix L, obtaining unique & optimal diagonal D which yields to good candidate for searching N cells out of $N^2$ HOL cells is critical. Obtaining weight matrix which indicates positional weight of HOL to be selected needs to addressed carefully.

Let A be any matrix then position weight of HOL at $A_{11}$ is associated with element $A_{22}$, $A_{33}$ and also with $A_{23}$, $A_{32}$ in a 3x3 matrix. As sequence $A_{11}$, $A_{22}$, $A_{33}$ is going to one of permuted diagonal, like wise $A_{11}$, $A_{23}$, $A_{32}$ is also another permuted diagonal which are candidate solutions. But sequence $A_{11}$, $A_{12}$, $A_{13}$ or $A_{11}$, $A_{21}$, $A_{31}$ are not at all participating in candidate solution of HOL at $A_{11}$. Selection of HOL at $A_{11}$ drops probability of selection of HOL corresponding to $A_{12}$, $A_{13}$ or $A_{21}$, $A_{31}$. It happens because of constraint that from one input port only one packet can be selected and only packet can be delivered to the output port. It indicates that positional weight of HOL can be obtained by cofactor elements only. Here we form weight matrix W from occupancy Queue-length matrix A.

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \Longrightarrow \begin{bmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \\ W_{31} & W_{32} & W_{33} \end{bmatrix}$$

Queue length matrix Weight Matrix

$W_{11} = A_{11}\ A_{22}\ A_{33} + A_{11}\ A_{23}\ A_{32}$
$W_{12} = A_{12}\ A_{21}\ A_{33} + A_{12}\ A_{23}\ A_{31}$
⋮ ⋮ ⋮
$W_{33} = A_{33}\ A_{22}\ A_{11} + A_{33}\ A_{21}\ A_{12}$ (4)

This process is nothing but obtaining partial determinant value at position $A_{ij}$. We have used this basic concept and modified Leibniz formula to obtain partial determinant value in this algorithm.
Hence

$$W_{ij} = \sum_{\sigma \in S_n} Per(\bar{\sigma}) \prod_{\substack{l=1 \\ k=i}}^{\substack{M=\begin{cases} k+N-1 & \le N \\ (k+N-1) mod N & >N \end{cases} \\ N}} A_{k,[j,\bar{\sigma}]_l} \quad (5)$$

Where, Per $(\bar{\sigma})$ = 1
$\bar{\sigma} = \sigma[m] = j + m$ if $j + m \le N$
$= (j + m) \bmod N$ if $j + m > N$
$\bar{\sigma}$ is vector of m elements where, m = 1, 2. . . N-1

In general, permutation $\bar{\sigma} = \{1,2,3\}$ and its permutations are,
Per$(\bar{\sigma})$ → (1,2,3) , (1,3,2) , (2,1,3) , (2,3,1) , (3,1,2) , (3,2,1)

Using above formula two sample positional weights are evaluated

**Example 1:**
**$W_{11}$ → i=1, j=1, N=3, m =1,2**
$\therefore \bar{\sigma} = \{1+1, 1+2\} = \{2,3\}$

$$W_{11} = \sum_{\sigma \in S_n} Per(2,3) \prod_{\substack{l=1 \\ k=1}}^{\substack{M=3 \\ N=3}} A_{k,[j,\bar{\sigma}]_l}$$

$$= Per(2,3) \prod_{\substack{l=1 \\ k=1}}^{\substack{M=3 \\ N=3}} A_{k,[1,2,3]_l} + Per(3,2) \prod_{\substack{l=1 \\ k=1}}^{\substack{M=3 \\ N=3}} A_{k,[1,3,2]_l}$$

$$= \prod_{\substack{l=1 \\ k=1}}^{\substack{M=3 \\ N=3}} A_{k,[1,2,3]_l} + \prod_{\substack{l=1 \\ k=1}}^{\substack{M=3 \\ N=3}} A_{k,[1,3,2]_l}$$

$$= A_{1,1}.A_{2,2}.A_{3,3} + A_{1,1}.A_{2,3}.A_{3,2}$$

$$= A_{1,1}\ (A_{2,2}.A_{3,3} + A_{2,3}.A_{3,2})$$

**Example 2:**
**$W_{23}$ → i=2, j=3, N=3, m= 4, 5**
$\therefore \bar{\sigma} = \{3+1, 3+2\} = \{4,5\}$ since both values are >N we have to take mod.
$\therefore \bar{\sigma} = \{4 \bmod 3, 5 \bmod 3\} = \{1,2\}$

$$W_{23} = \sum_{\sigma \in S_n} Per(1,2) \prod_{\substack{l=1 \\ k=2}}^{\substack{M=1 \\ N=3}} A_{k,[j,\bar{\sigma}]_l}$$

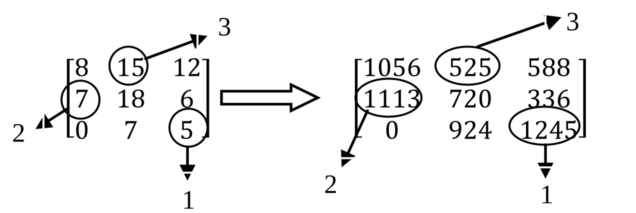
$$= Per(1,2) \prod_{\substack{l=1 \\ k=2}}^{\substack{M=1 \\ N=3}} A_{k,[3,1,2]_l} + Per(2,1) \prod_{\substack{l=1 \\ k=2}}^{\substack{M=1 \\ N=3}} A_{k,[3,2,1]_l}$$

$$= \prod_{\substack{l=1 \\ k=2}}^{\substack{M=1 \\ N=3}} A_{k,[3,1,2]_l} + \prod_{\substack{l=1 \\ k=2}}^{\substack{M=1 \\ N=3}} A_{k,[3,2,1]_l}$$

$$= A_{2,3}.A_{3,1}.A_{1,2} + A_{2,3}.A_{3,2}.A_{1,1}$$

$$= A_{2,3}\left(A_{3,1}.A_{1,2} + A_{3,2}.A_{1,1}\right)$$

Now consider the queue-length matrix L and its associated weight matrix W given below.




Weight matrix obtained, provides solution with greedy approach of selection to maximum weight i.e. “1245” positional weight is highest corresponding to HOL [3,3] which is selected first. In second selection “1113” positional weight is selected i.e. HOL [2,1] is selected. Hence HOL [3,3] [2,1] and [1,2] are selected. There are no multiple solutions observed in this PDMWM policy.

## 4. Performance Metrics:

Our attempt is to provide optimal solution out of 'N!' solutions. Obtaining weight matrix is computationally complex and can be achieved by using multicore processor or hardware parallelization. Our attempt is not to suggest any hardware implementation but one way to provide path to get optimal solution. As we had seen Queue-length matrix L has overall average queue-length at time t

$$\bar{L}(t) = \frac{\sum_{ij} L_{ij}(t)}{N^2}$$

Let selected diagonal be 'D' after implementing any scheduling policy.

Let variance of optimal diagonal solution be $\sigma_D^2$

$$\sigma_D^2(t) = \frac{1}{N} \sum_{ij} [L_{ij}^D(t) - \bar{L}(t)]^2$$

Let average queue-length of optimal diagonal 'D' is

$$\overline{L_D}(t) = \frac{1}{N} \sum_{ij} L_{ij}^D(t)$$

Table 1: Result Summary

| Switch Size / Policy | Cells Switched | $\bar{L}$ | $\sigma_D^2$ | $L_D$ | $L_D.\sigma_D^2$ |
|---|---|---|---|---|---|
| LQPS | 2 | 8.66 | 57.79 | 10 | 577.9 |
| LQPSP | 2 | 8.66 | 33.68 | 10.33 | 347.9 |
| WMAXP 1 | 3 | 8.66 | 33.68 | 10.33 | 347.9 |
| 2. | 3 | 8.66 | 5.55 | 8.66 | 48.06 |
| 3. | 3 | 8.66 | 3.422 | 7.0 | 23.94 |
| 4. | 3 | 8.66 | 18.78 | 9.0 | 169.02 |
| PDMWM | 3 | 8.66 | 18.78 | 9.0 | 169.02 |

With reference to above table LQPS is discarded as it switches less number of packets than N. Select the policies with average Queue-length of optimal diagonal, higher than overall average queue- length $\bar{L}(t)$ with minimum variance $\sigma_D^2(t)$.

If deviation of Queue-length of optimal diagonal is minimum with reference to overall average Queue-length $\bar{L}$ it indicates that queue-lengths of selected diagonal are uniformly distributed. If you choose optimal diagonal with average diagonal queue-length $L_D$ is higher than overall average queue-length $\bar{L}$ and variance $\sigma_D^2$ is high then queues chosen are having non-uniform variation in queues or queues selected will be dominated by hot-spot traffic.

## 5. Conclusion

PDMWM is providing unique and optimal solution for packet selection policies in VOQ switches. It's computationally complex but gives optimal solution. This is tested for various random queue occupancy matrix and observed that it improves the throughput delay performance. This is theoretical attempt to show and suggest one method of MWM algorithm.

## 6. Future Scope

PDMWM scheme stability analysis evaluation with obtaining fast solution under higher size of switch is real challenge.

## References

[1] N.McKeown, A. Mekkittikul, V.Anantharam, J.Walrand "Achieving 100% throughput in Input output queued Switch", IEEE Trans. On Communication, Vol.47, No.2, pp.188-201, April 1999.

[2] Karol M., Hluchyj M., Morgan S., "Input versus Output Queuing on a Space Division Switch", IEEE Transaction on Communications. Vol.35, n.12, p.1347-1356, Dec. 1987.

[3] Awdeh, R.Y. and H.T. Mouftah, "Survey of ATM Switch Architectures", Computer Networks and ISDN Systems, vol. 27 (1995), pp. 1567-1613.

[4] Hui, J.Y. and E. Arthurs, "A Broadband Packet Switch for Integrated Trans- port", IEEE J. Sel. Areas Commun., vol. 5, no. 8, Oct. 1987, pp. 1264-1273.

[5] Pankaj Gupta, "Scheduling in input queued switches: A survey", 1998

[6] M.Ajmone Marsan, A.Bianco, E.Filippi, "A Comparison of Input Queuing Cell Switch Architectures", EEE BSS'99, 3rd International Workshop on Broadband Switching Systems 1999.

[7] Yanping Gao, Christoforos Kachris, Manolis Katevenis "An Efficient Sequential Iterative Matching Algorithm for CIOQ Switches" Computers and Communications (ISCC), 2011 IEEE Symposium, 2011, pp. 558-563

[8] Bing Hu, Kwan L. Yeung, and Zhaoyang Zhang, "Minimizing the Communication Overhead of Iterative Scheduling Algorithms for Input-queued Switches", Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE, 2011, pp. 1-5

[9] S. N. Kore, V.B. Dharmadhikari, H. S. Jamadagni, "Cell Selection Policies in Multiple Input Queued ATM Switch", CCBR'99

**Sunil Kore** received his B.E degree in Electronics and Telecommunication from University of Pune at Pune and M.E. degree in electronics from Shivaji University at Kolhapur in 1983 and 1991respectively.
He has worked as scientific officer at Tata Institute of Fundamental Research (TIFR) at Pune from 2nd Nov'83 to 1st Oct'84. Currently he is an Associate Professor of electronics engineering at Walchand College of engineering, Sangli, Maharashtra.

**Ajinkya Biradar** received his B.E. degree in Electronics and Telecommunication from University of Pune at Pune in 2010 and is pursuing M.Tech in electronics from Walchand College of engineering, Sangli

**Sayali Kore** received her B.Tech degree in Electronics through Autonomous institute Walchand College of engineering, Sangli, Maharashtra (India) in June 2012. She is Gold medalist among the students of B.Tech in Electronics Engineering.