# A Novel Dynamically Optimized Embedded Video Burst Scheduler to Enhance the System QoS

**B. Bag, A. K. Jana, M. K. Pandit**

***Abstract*— Real-time video processing is still now a formidable task for the strict requirement on latency control and packet loss minimization. Burst processing has come to the rescue by offering buffer less operation and separation of control and data information. In this paper a novel dynamically-optimized embedded burst scheduling method suitable for processing class-differentiated video channels has been proposed. The method is based on statistical Markov chains where the initial scheduled Markov transition probabilities are subsequently adaptively reconfigured by the central scheduler to maintain the best system Quality of Service (QoS).***

***Keywords*—Embedded systems, QoS, Markov process, Reconfigurable computing, Video communication.***

## I. INTRODUCTION

In modern days the Internet is seen to be the most promising approach to build the next generation communication infrastructure. With the increasing demand for information intensive multimedia applications which require dynamic scheduling and quality of service (QoS) provisioning is becoming more and more important. An efficient and dynamic scheme is crucial to guarantee the QoS and to optimize the scheduler specially for mission-critical and real-time applications like video processing, that requires low latency as well as low packet losses.

QoS for embedded video scheduling systems for an individual process has been reported by Yoo et. al., [1]. S. J. Yoo [2] suggested some new approaches towards scheduling without QoS awareness. Safavian [3] and Tomoyoshi et. al., [4] have studied QoS in embedded video processing systems but none of them considered Markov chains. The concept of dynamically-optimized system-QoS based on Markov statistical video scheduler has not yet been reported. In this paper the authors used the concept of dynamically-optimized system-QoS based on Markov statistical video scheduler. This QoS provisioning method can adapt the changes in the traffic pattern. In this paper priority-based incoming traffic rate control for service differentiation and congestion control in sending real time traffic to the processor in a video processing environment is used.

Most of the earlier works [3-6] on finite state modeling of schedulers considered the four types of states for each process; they are- sleeping, ready, executing and blocked. The reported results [3-6] are not compatible with dynamically optimized system QoS. In this paper the authors have proposed and modeled each of the processes in a video processing environment as an individual Markov process which is compatible to dynamically optimized system QoS.

QoS scheme based on burst in embedded video processors for real time applications to minimize the system loss probability has been proposed in this paper. The proposed scheme used the offset time instead of buffer to isolate data and control bursts. QoS is enhanced by taking the upper bound of the system loss probability by optimizing the parameters values. The global interest for real time multimedia applications such as VoD and IPTV had increased to a large extent in last few years. , Servetto et.al. [10] stated TCP is not well suited for real time data transmission due to following characteristics:

- TCP built in transmission mechanism
- TCP strict congestion control mechanism
- TCP does not work suitably for real time application

On the other hand UDP (User Datagram Protocol) also does not provide adequate support for real time data transmission as it suffers from low QoS. Therefore RTP (Real Time Protocol) was designed. RTP along with its associated profiles and payload formats provide a framework for real-time media transmission. RTCP (Real Time Control Protocol) provides participant identification, media stream synchronization and reception quality feedback when it is running alongside RTP. We propose here a video scheduling mechanism that can take decisions at runtime for QoS provisioning and increase in throughput. Optimization and reconfiguration techniques work by analyzing the traffic pattern with the help of Feedback control system of the proposed scheme.

## II. SYSTEM OVERVIEW

In our system we differentiate between classes of the incoming traffic channels. The traffic is assumed to be of three different classes having different priority and traffic pattern.

Manuscript received August 22, 2011.

**B. Bag,** Department of Electronics and Communication Engineering, Haldia Institute of Technology, Haldia, West Bengal, India (E-Mail: bani305@gmail.com).

**A. K. Jana,** Department of Electronics and Communication Engineering, Haldia Institute of Technology, Haldia, West Bengal, India (E-Mail: asimkjana@gmail.com).

**M. K. Pandit**, Department of Electronics and Communication Engineering, Haldia Institute of Technology, Haldia, West Bengal, India (E-Mail: mkp10011@yahoo.com).

Each traffic class is served by a unique process in the embedded processor. One process is broken down into several threads (k). Traffic comes in burst with control and data (bursts) separated by **offset time $t_{off}$**, as shown in Figure 1. A burst is composed of a set of fixed-length segments of the same traffic class. We assume that the input traffics are in a normalized form.

The offset time is required for data and control isolation. Lower bound on the burst loss probability is analyzed. Simulations are also conducted to evaluate the QoS performance in terms of burst loss probability. We have shown that moderate amount of buffer can enhance the QoS very efficiently.

## III. METHODOLOGY

In our system we assume there are three process of different priority levels based on their traffic rates and required service rates such that the waiting time is negligible or very less. High priority is treated as class 1, medium priority is taken to be class2 and low priority class 3. To serve the incoming burst depending on the traffic rate we apply the concept of software threads to work independently. The maximum number of threads allocated to each process is proportional to its priority level. Thus traffic intensity (load) normalized to number of threads is same for all processes.

Although we do not intend to use buffer but using of small amount of buffer **enhances** the QoS **remarkably.** As to the process of highest priority, it does not wait at all or wait for minimal time for processor availability so for this process there is no need of buffer or very small amount of buffer is sufficient. We take into account that in real time process with the highest priority process should not lose any burst at any traffic condition. The lower priority process should wait till the highest priority process is executed and processor is free to serve the lower priority process. Consequently the lower priority process context needs buffer storage. In our system we have implemented the preemptive scheme with priority scheduling that is the lower **priority** process can be **pre-empted** by the **higher priority process**. So it is clear that class 3 traffic bursts will have a relatively high loss probability. Considering each process as an **FSM** state with nondeterministic operations separating each state. A large switch statement examines state variables on each invocation to determine which case to execute.

### A. Video Burst

The burst length should neither be very small nor be very large. If the burst length is small, throughput will be less, while if the burst length is very large, delay will grow up. This is due to higher wait times suffered when packets are formed into larger bursts - this causes additional overall delay. The burst length should therefore be optimized for the best QoS.

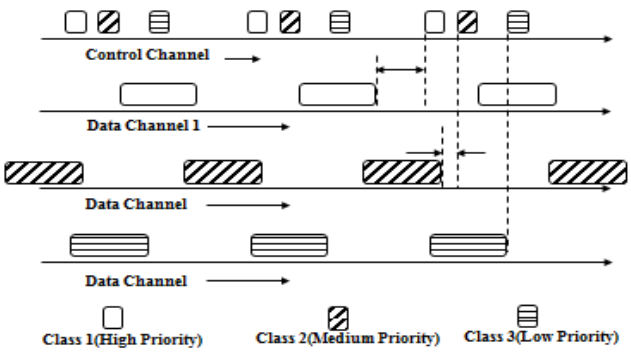


**Fig. 1 Video Channel Bursts with assigned priorities**

### B. Markov Process

The video bursts are assumed to arrive with a **Poisson** probability distribution. The high-priority classes have a longer offset time than the low-priority classes as shown in Fig. 1. For this system, we calculate the system burst loss probability and the burst loss probability for each priority class. We use **three** types of **video traffic** to evaluate the performance of the **embedded video processor**. The first one is NTSC **hard real-time** video signal having extremely tight timing requirements and so treated as class I. Second one is **real-time** video having slightly less strict timing requirements and considered as class II traffic. Third one is streaming video signal and assigned class III priority. Highest priority is given to NTSC video signal processing where error detection and error correction algorithm are incorporated. The processor allocates maximum number of threads to this process. Consequently, the processor will be busy most of the time to serve the first process. We develop here a **real-time scheduling algorithm** for video processors to enable multiple video application run concurrently with no or low latency.

The prime focus in this paper is made on how to achieve high **QoS** in hard real time systems for embedded applications. Embedded systems have multiple concurrent activities that must meet their deadlines for maintenance of high QoS. The problem is solved by implementing software threads that trims the system cost by eliminating dedicated hardware peripherals. Further, they enhance system flexibility.

### C. Optimization Parameters

We analyze the lower bounds on the burst loss probability and **minimize** it by **optimizing** the process (state) to process (state) **transition probabilities** in a **novel** way. The results obtained from analysis and simulations are described. The number of average **burst-length** (L) and the number of **threads** (k) for different classes are considered as the key design parameters in the system. The class I channel generates bursts according to an exponentially distributed arrival rate with average $\lambda_i$ and with an exponentially distributed service rate with average $\mu_i$ where $\mu_i = 1/L_i$. The normalized traffic intensity of class i is $\rho_i = (\lambda_i / \mu_i)/k = r_i / k$ where traffic intensity $r_i = \lambda_i / \mu_i$ .

The processes in the embedded processor are modeled as Markovian. The lower bound on loss probability is determined by assuming that there is **minimal buffer**. It is given according to **Markov** model as

$$P_{bl} = \frac{r^D}{k^{D-k}} P_0 \qquad (1)$$

$$\text{Where } P_0 = \left[ \sum_{n=0}^{k-1} \frac{r^n}{k!} + \sum_{n=k}^{D} \frac{r^n}{k^{n-k}.k!} \right]^{-1} \qquad (2)$$

Where the total number of bursts that can be processed by the embedded processor is D = k + k.N, N = physical memory block size allocated to each thread.

Currently we consider our system to work with only a single processor and further work will be done for multiprocessor systems. The performance of this model is assessed by evaluating the burst loss probability. The processing nodes are assumed to follow Poisson traffic model. The results of burst loss probability vs **normalized** traffic Intensity for all classes using equation (2) have been shown in Fig 2.

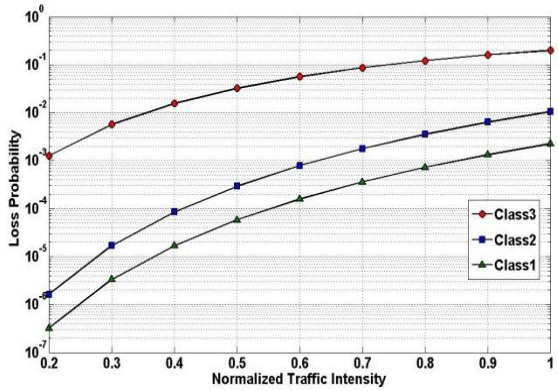


**Fig. 2 Loss probabilities for different classes of traffic**

We observe that class III traffic suffers the highest loss and as the traffic escalates loss probability also rises. From Figure 2 it is clear that up to 0.3 unit traffic loads, class III burst loss probability grows rapidly and at 0.3 units, it is approximately $5\times10^{-3}$. Beyond 0.3 unit of traffic load, burst loss probability rises at a slower pace, steadying at approx. 0.2 to 0.3 beyond a load of 0.8.

### D.System QoS

The system QoS is calculated using **Markov chain** analysis where the different processes are treated for their state transition probabilities. In our model we consider three processes (which are adequate for many low-end embedded applications) assigned to a single processor. Considering a finite state machine (FSM), the three states $<P_1>$, $<P_2>$, $<P_3>$ define the execution of three corresponding processes by the CPU. As in a single processor system the CPU executes only one process at a time, so in order to execute several processes, transition from one process to another via **context switching** is essential. Context switching is to be planned properly because it requires CPU overhead. Consequently it cannot be allowed at arbitrary times. It must be kept to a minimum by proper allocation of the state transition probabilities $p_{ij}$.

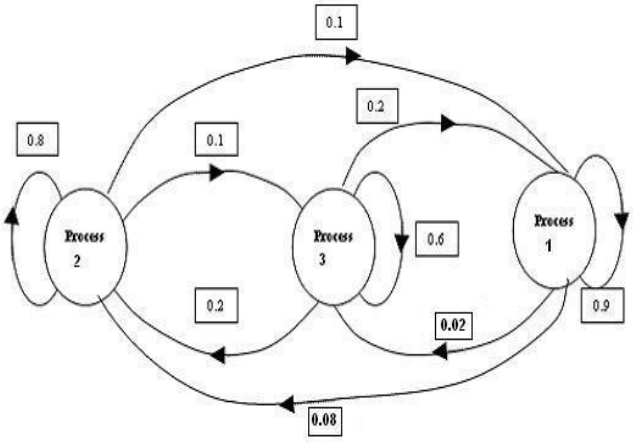


**Fig. 3 State transition model for the three processes**

Fig. 3 demonstrates the three-process state transition diagram. The directed edges determine CPU's transition or switching from one process to another. The CPU may stay at its present state or switch over to any one of the other two states, determined by external **events**. The weights of the corresponding edges are considered as the transition probabilities.

## IV. SIMULATION MODEL

For the sake of simulation, we consider two matrices 1) P1, a state transition matrix for Markov model assumed (here three state model and 2) E, an error (loss) probability matrix (at **normalized** traffic intensity of 0.6).

$$P_1 = \begin{bmatrix} 0.90 & 0.08 & 0.02 \\ 0.10 & 0.80 & 0.10 \\ 0.20 & 0.20 & 0.60 \end{bmatrix} \qquad (1)$$

$$E = \begin{bmatrix} 0.9998 & 0.9990 & 0.9500 \\ 0.0002 & 0.0010 & 0.0500 \end{bmatrix} \qquad (2)$$

The simulation is done by MATLAB version 7.0. We perform simulation for calculating error vector which gives error positions in 20000 sequences (iterations) considering random numbers, and the state sequence matrix which deals with state transitions. The probability of finding the processor in a given state can be calculated from the state sequence matrix [8]. Similarly error probability can be found from the error vector.

At normalized traffic intensity of 0.6, the probabilities of processes 1, 2, and 3 are found to be 0.59, 0.24 and 0.17 respectively and the system (loss) error probability 0.03. It is seen from Fig 4 that the loss probability grows up with traffic. Consequently it keeps changing with varying **traffic pattern**.

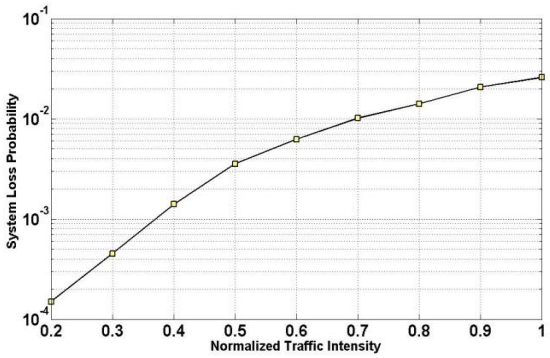


**Fig. 4 The system loss probability as a function of normalized traffic intensity**

## V. STATIC OPTIMIZATION

We consider two other options for the transition matrix with the transition probabilities inter-changed as shown in

$$P_2 = \begin{bmatrix} 0.80 & 0.10 & 0.10 \\ 0.08 & 0.90 & 0.02 \\ 0.20 & 0.20 & 0.60 \end{bmatrix} \quad (3)$$

$$P_3 = \begin{bmatrix} 0.90 & 0.02 & 0.08 \\ 0.20 & 0.60 & 0.20 \\ 0.10 & 0.10 & 0.80 \end{bmatrix} \quad (4)$$

Equation 4 and compare their overall system loss probabilities in Fig 5. Matrix P1 is finally **selected** as it is seen to be the most **optimum**, giving **minimum** system error.

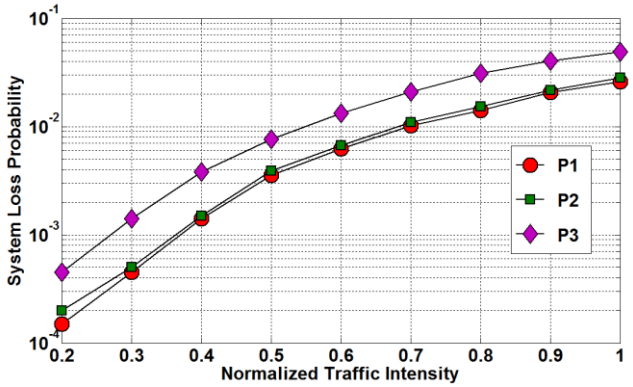


**Fig. 5 System loss probabilities for different P matrices**

## VI. ANALYSIS OF SYSTEM LOSS PROBABILITY

Every **data burst** is preceded by the **control burst**, which is sent ahead to make the processor aware that the data burst is on the way. The control burst contains the following parameters of the data burst: expected arrival time, size of the burst, class of traffic. The performance metrics are the burst loss probability and are queuing delay as a function of traffic intensity, the maximum delay time, the offset time difference, the number of classes.

The offset time must be carefully designed for each channel to allow enough time for the burst to be processed. High priority traffic is given an extra offset time to reduce its blocking probability. We simulate and analyze the system loss probability for particular values of burst size and offset-time. Keeping all other parameters same, the system loss probability for various offset time differences are plotted in Fig. 6. As is clearly expected the system loss probability increases with traffic intensity. Further, when the offset time is lower, the loss is higher as the processor gets more time to serve the incoming bursts. From Fig. 6 it is clear when the traffic intensity is too large beyond 0.9; the loss probability for different offset-time is tending to merge closely with each other.

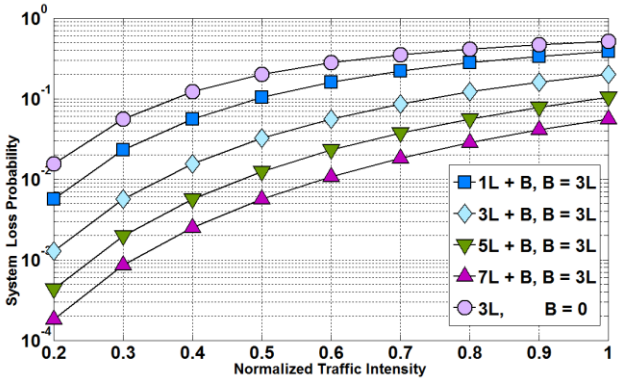


**Fig. 6 System loss probabilities for various offset time**

Though a long maximum delay (end-to-end) time can reduce the loss probability, one needs to keep it as short as possible; otherwise the latency will be a great deal poor. In this subsection, we show the effect of maximum delay time on QoS performance such as burst loss probability under different traffic load. Fig. 7 shows the loss probability as a function of the maximum delay time. The parameters are taken as n = 4, k=2, N=3(d=6), ρ=0.8, $t_{diff}$ = 3L + B

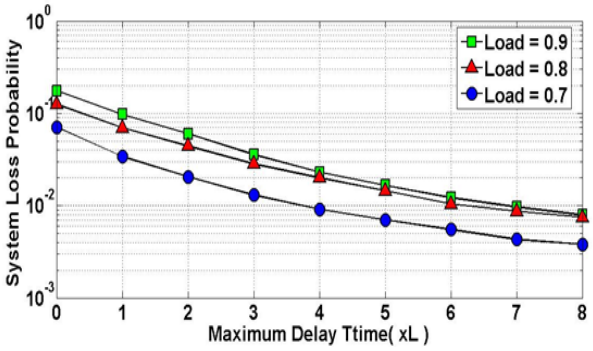


**Fig. 7 Burst loss probability under different traffic load**

As is obviously expected Fig. 7 demonstrates that system loss probability increases with the higher traffic load.

## VII. DYNAMIC OPTIMIZATION AND RECONFIGURABILITY

As the traffic pattern changes, the pre-allocated state transition probabilities of matrix P1 are **unable** to maintain the QoS at its **maximum**. We solve this by **re-configuring** matrix P1 with the help of reconfiguration.

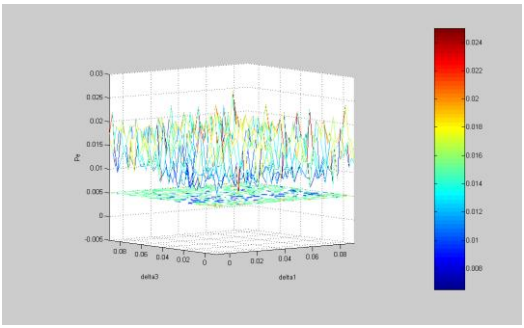

**Fig. 8 Probability of error vs Δ1, Δ3 (Δ2 = 0)**

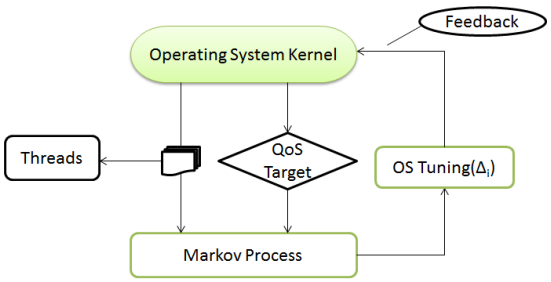


**Fig. 9 Feedback control system for our scheduler parameters Δ1, Δ2 and Δ3 as shown in Equation 5.**

$$P_{Reconfig} = \begin{bmatrix} 0.90-2\Delta_1 & 0.08+\Delta_1 & 0.02+\Delta_1 \\ 0.10+\Delta_2 & 0.80-2\Delta_2 & 0.10+\Delta_2 \\ 0.20+\Delta_3 & 0.20+\Delta_3 & 0.60-2\Delta_3 \end{bmatrix} \quad (5)$$

This brings the probability of error to a minimum 0.007 and hence QoS is back to **permissible maximum** as shown in Fig 8. The reconfiguration is implemented by a feedback controller as shown in Fig. 9

## VIII. CONCLUSION

We have proposed and designed a novel dynamically-optimized scheduler using Markov statistics. We have also shown the optimal burst blocking probability which can be maintained over a range of traffic loads by dynamically adjusting scheduler priority levels. The proposed scheme gives us very low error probability (0.007) after upper bound of error probability taking into consideration. Due to very low error probability of the proposed scheme, it is well suited to the embedded systems to maximize the performance.

## REFERENCES

1. M. Yoo, C. Qiao, and S Dixit, "QoS performance of optical burst switching in IP-over-WDM networks", IEEE/OSA J. Light wave Technology, Vol. 18, No. 10, Feb. 2000, pp. 2062-2071.
2. S. J. Ben Yoo, "Optical packet and burst switching for the future photonic internet", IEEE/OSA J. Light wave Technology, Vol. 24, No. 12, Dec. 2006, pp. 4468-4492.
3. S. Rasoul Safavian, "How to dimension wireless networks for packet data services with guaranteed QoS", Bechtel Telecommunications Technical Journal, Vol. 3, No. 1, March17-19, 2008
4. S. Tomoyoshi, T. Kosuke, "Table-based QoS control for embedded real-time systems", Proc of the ACM SIGPLAN 1999 workshop on Languages, compilers, and tools for embedded systems, 1999, pp. 65-72.
5. R. Jain, C. J. Hughes and S. V. Adve, "Soft Real-time Scheduling on Simultaneous multithreaded processors", Proc of 23rd Real-time systems symposium (RTSS-23), IEEE Press, 2002, pp. 134-135.
6. A. Snaverly, D. M. Tullsen and G. Voelker, "Symbiotic job scheduling with priorities for a simultaneous multithreaded processor", Proc of 9th International Conf on Architectural Support for programming languages and operating systems (ASPLOS-9), ACM Press, 2000, pp. 234-244.
7. S. Addepallil, P. Andersen and G. L. Barnes, "Efficient Resource Matching in Heterogeneous Grid Using Resource Vector", Int Journal of Computer Science and Information Technology, Vol. 2, No. 3, June 2010, pp. 1-10.
8. G. Wang, "Efficient Resource Matching in Heterogeneous Grid Using Resource Vector", IEEE Transaction on Aerospace and Electronic Systems, Vol. 46, No. 3, July 2010, pp. 1492-1502.
9. K. S. Chan, Kwan. L. Yeung "Performance Analysis of Burst Segmentation Schemes supporting Multiple Traffic", IEEE No. 0-7803-8924-7, May 2005, pp. 495-499.
10. S. D. Servetto, K. Nahrstedt "Video Streaming Over the Public Internet: Multiple Description Codes And adaptive Transport Protocols", 1999 IEEE No. 0-7803-5467-2/99, pp. 85-89.
11. M. F. Ngatman, Md. A. Ngadi, J. M. Sharif "Comprehensive Study of Transmission Techniques for Reducing Packet Loss and Delay in Multimedia over IP", IJCSNS, VOL-8 No. 3, March 2008, pp-292-299
12. Jian Li, Ye-Qiong Song "DLB: a novel real-time QoS control mechanism for multimedia transmission", International Journal of High Performance Computing and Networking 2009 - Vol. 6, No.1 pp. 4 - 14

## AUTHORS PROFILE

**Banibrata Bag,** Received the B.E. from Burdwan University in 2004 and M.Tech under WBUT (*West Bengal University of Technology*) in 2009. He has worked as a software engineering (from 2005 to 2007), and currently working as an assistant professor (from 2010) in the Dept. of Electronic and Communication Engineering, Haldia Institute of Technology, Haldia, WB, India. His research interest includes embedded system and processor architecture for multimedia application. He is a professional member of ACM (*Association for Computing Machinery*).

**Asim K. Jana,** Asim K. Jana received his bachelor's and master's degree in electronics Engg from Electronics and Telecom Engg Dept, Jadavpur University, India in 1988 and 1995, respectively. He has 10 years of industrial experience in the domain of embedded systems. Currently he is an associate professor on computer systems. He has 10 international research papers.

**M. K. Pandit,** Malay K Pandit received his B.E and M. E degrees in Electronics Engineering from Electronics and Telecom Engg Dept, Jadavpur University, India in 1989 and 1991, respectively. He received his PhD from UK's renowned Cambridge University in 1996. He did his post-doc from the Optoelectronics Research Centre, City University of Hong Kong till 2002 where he pioneered the use of polymers for optical waveguide applications. He then took a corporate career where he worked in a fiber optic company "FONS (I) Ltd" in the domain of optical networking. Now he is the Dean of School of Engineering and the Professor in the Electronics Engg Dept of the Haldia Institute of Technology where he focuses on embedded systems, including their usage in WDM optical networks. He has 45 international publications in this area. Dr. Pandit was awarded the National Scholarship of the Government of India in 1983 and the Nehru Scholarship of the Government of India during his Ph.D. studies.