A QoS-Enhanced Stochastic Scheduler for Smart Routers with Timing Jitter Analysis

Suman Paul[1,2], Malay Kumar Pandit[1]

[1] Department of Electronics and Communication Engineering, Haldia Institute of Technology,
Maulana Abul Kalam Azad University of Technology West Bengal, India
[2] School of Engineering and Technology, Maulana Abul Kalam Azad University of Technology West Bengal, India
paulsuman999@gmail.com, mkpandit.seci@gmail.com

Abstract

In this paper we propose and analyze performance of a utilization-driven QoS-enhanced intelligent stochastic fair packet scheduler for IP traffic in a smart router with timing jitter analysis. The unique advantages of the proposed scheduler are: the scheduler offers the benefit of arbitrarily pre-programming the process utilization ratio; it solves the problem of starvation for low priority processes; it solves the major bottleneck of the benchmark scheduler Earliest Deadline First's (EDF) failure at heavy loads. We consider three classes of multimedia IP traffic: VoIP, IPTV and HTTP. We analyse the performance of the scheduler addressing important QoS parameters, scheduling jitter, the scheduler's noise response, packet loss rate and mean waiting time. Simulation results show that the proposed scheduler has a performance improvement compared to current state-of-the-art schedulers.

Keywords: Hidden markov model (HMM), Packet loss rate (PLR), Quality of service (QoS), Scheduling jitter

1 Introduction

The overall quality of a network connection has a significant impact on the performance of network applications. Improving the quality of service in IP networks is a major challenge for real-time communication. Quality of Service (QoS) [1] in telecommunication systems is directly related to the network performance of the underlying routing systems. A router is a specific case of soft-real time embedded systems. Scheduling of IP traffic is an integral part of modern IP routers. QoS is of prime concern in designing modern routers as it addresses key attributes and statistical representation of network behavior parameters, like packet loss rate (PLR) [2], jitter (delay variation) [3], latencies, sources of errors, resource availabilities, end-to-end delay, fair bandwidth allocation etc. We present here a probabilistic framework for a novel optimal intelligent embedded computing scheduler, QUEST (quality-of-service enhanced stochastic), for smart IP routers with timing jitter analysis. QUEST is strictly traffic class-sensitive and fully aware of SLAs and it is a deadline-aware utilization- driven scheduler. We identify two major gaps in scheduler research, the starvation of low priority processes and poor performance of the premier EDF scheduler and its related scheduling algorithms as the rise of traffic mean waiting time to an unacceptably high levels at heavy traffic loads. EDF is a dynamic priority based real-time scheduling algorithm which dynamically assigns priorities to each process based on its absolute deadline. The process with the earliest absolute deadline at any given time has the highest priority of all processes.

1.1 Scheduling Characteristic

The proposed QoS-enhanced intelligent stochastic packet scheduler, QUEST, for IP routers is based on pre-emptive scheduling but it differs from the conventional schedulers in that it is probabilistic in nature in order to keep the utilization fixed in a fair way and offers the following advantages:

(i) An optimum utilization close to 100 percent is enforced. In this scheduling scheme, process utilization, U_i for a process P_i, is expressed as,

$$U_i = \frac{T_i}{D_i} \quad \textbf{(1)}$$

where T_i is the fraction of time spent for execution of process P_i. D_i is denoted as the deadline of the process P_i. The state probability vector of process utilization ratio of n number of class processes running in a system can be expressed as, $\prod$

$$\prod = [U_1 : U_2 : .. U_{n-1} : U_n] \quad \textbf{(2)}$$

(ii) A machine-learning feedback controller is used to in QUEST to implement the adaptability and re-configurability. This feedback-controller with the help

---

*Corresponding Author: Suman Paul; E-mail: paulsuman999@gmail.com

DOI: 10.3966/160792642018121907026

of error feedbacks learns and takes corrective decisions to maximize the system QoS.

(i) Higher priority processes (traffic) cannot monopolize the processor and the lower priority processes acquire a guaranteed minimum amount of processor time.

In practice, for an end-to-end QoS sensitive traffic, which has a commitment to deliver on time, the process utilization for different classes of traffic is tailored in such a manner that a guaranteed minimum amount of processor attention for each traffic is maintained. For multimedia embedded (router) applications considered in this paper, Voice over Internet Protocol (VoIP) [4], Internet Protocol Television (IPTV) [5] which are real-time traffic and web browsing using Hyper Text Transfer Protocol (HTTP) which is a best effort network traffic processes follow a long-tailed *Pareto* distribution of process utilization ratio. In this proposed scheduling model, a *target* practical process utilization ratio of 80:16:4 has been is achieved and maintained as per designer's requirement.

1.2 System Quality of Service (QoS)

The PLR encountered in system activities that may arise due to different errors like deadline miss, cache misses (L1 and L2), page fault, etc. Practical cache miss error probabilities come in the range of [10^{-2}-10^{-1}]. Practical deadline miss error probabilities come in the range of [0.013-0.12]. Acceptable practical jitter as recommended by different vendors in a network are in the range of 0.5 ms [6]. For practical real-time tasks, the deadline varies in the range of 10-300 ms.

2 Literature Survey

In routers, the simplest First-come first-served (FCFS) scheduler cannot differentiate traffic classes. The authors in [7], have developed bounds for Deadline Minus Jitter Monotonic (DMJM) and Earliest Deadline First (EDF) message scheduling techniques. Kim [8] have proposed an enhanced timing recovery algorithm combined with active jitter estimation to improve voice quality. The proposed algorithm overcomes the effect of transmission jitter by way of expanding or compressing each packet according to the estimated predicted network delay and variations more accurately than conventional algorithms. In [9], the authors have proposed a queuing delay control and adjustment method. The QoS parameter defined in terms of per-service traffic flow for real-time multi-service traffic. This method shows how to control the queuing delay value at the specified waiting delay by adjusting the traffic arrival probability resulting the QoS delay requirement where real-time services may be guaranteed.

Based on literature survey it is observed that in a multitasking scheduler in IP routers, dynamically optimizing the system QoS based on Markov chain model has not been specifically focused. Both in the static priority scheduling: rate monotonic (RM) and dynamic priority scheduling: earliest deadline first (EDF) and its variants, the lower priority processes have to wait due suspension of execution by the higher priority traffic. In a dynamic environment of real-world applications for an overloaded system using EDF, the processes miss deadlines frequently resulting in very low value of throughput. In EDF all network traffic classes receive the same miss rate irrespective of deadline requirements and EDF scheduler does not honour class differentiation for traffic and therefore fails to comply with the service level agreements (SLAs) with client processes. Last, EDF and its variant A-EDF are deadline driven, where process utilization has no clear focus.

3 Proposed System Model

TheThe underlying model behind this scheduling framework is a Hidden Markov Model (HMM) [10]. The design has been implemented for three class processes - VoIP, IPTV and HTTP. Each process in this scheme modelled as a particular Markov state. The processes settle to a steady state probability distribution according to time evolution.

Since HMM is an NP-Hard problem Markov initial TPM parameters (matrix elements) are calculated *apriori* using machine learning Metropolis-Hastings algorithm. Metropolis-Hastings algorithm is a special class of Markov Chain Monte Carlo (MCMC) method, with constraints like the diagonal elements of the TPM are in the range: [0.4-0.9] and the non-diagonal elements arein the range: [0.01-0.6]. It has been observed that a faster convergence is achieved in such cases. In real-time network applications where meeting a deadline is a crucial issue, the need arises for reaching the desired steady state equilibrium in a fast way. Hence Metropolis-Hastings algorithm is the preferred option when the Markov chain converges quickly to the equilibrium distribution. Because of Markovian property, target steady state probability distribution can be generated. The corresponding TPM is estimated by maximum likelihood.

For multimedia IP traffic considered in this work, the desired (fractal Pareto type) steady-state distributions are of the order of 0.80: 0.16: 0.04 as justified later in Section 4 with Table 1. So ξ=0.8, φ=0.16 and ï=0.04 are considered. An initial approximate estimate for the 3×3 Transition Probability Matrix (TPM), *'T'* is estimated by using the *machine learning* Metropolis-Hastings algorithm to provision a steady state distribution of process utilization ratio 0.80: 0.16: 0.04. '*T*' is stated in Eq. (3).

Table 1. Service models parameters

Traffic class	Service type	Deadline (ms)	Arrival feature
P_1 (VoIP)	RT-GBR	(ITU G.711)20	MMPP
P_2 (IPTV)	non-GBR	(ITU G.114) 100	MMPP
P_3 (HTTP)	Best effort non-GBR	400	MMPP

$$T = \begin{pmatrix} 0.90 & 0.08 & 0.02 \\ 0.39 & 0.56 & 0.05 \\ 0.42 & 0.18 & 0.40 \end{pmatrix} \quad \textbf{(3)}$$

The Π, the state probability vector, is treated as process utilization ratio. Ignoring apriori information, an initial unbiased state probability vector, $\Pi_0 = 1/3[1\ \ 1\ \ 1]$ is applied and the estimated final state probability vector, Π_f is obtained as, [0.79829:0.16154: 0.040071]. We further apply an initial biased state probability vector, $\Pi_0 = [0.1\ 0.5\ 0.4]$.

The estimated final state probability vector, Π_f is obtained as, Π_f = [0.79837:0.16155:0.040075]. These two vales of Π_f are approximately same. The result *confirms* that a final practical process utilization ratio, $\Pi_u = [U_1 : U_2 : U_3]$ distribution i.e. [0.80: 0.16: 0.04] for three processes, has been achieved, irrespective of the initial distribution. It is to be noted that a specific value of U_i, achieved here, is under the control of designer's choice. In general, any target v*alues* of Π_f, namely, *[0.81 0.130.06], [0.65 0.25 0.10]*, etc. can be achieved as per designer's requirement because Metropolis-Hastings algorithm can generate any arbitrary desired steady state distribution

4 Scheduling Framework

The multi-service packet scheduling framework QUEST, as shown in Figure 1accepts three different classes of incoming multimedia traffic - VoIP, IPTV and HTTP. Traffics streams are classified by a classifier and fed to three distributed FIFO queues: Q_1, Q_2 and Q_3 for VoIP, IPTV and HTTP, respectively. The proposed model is defined as M/BP/1/./QUEST. In this underlying model, 'M' denotes traffic arrivals which are of Markovian type modulated by Poisson process (MMPP). According to recent approaches, for a settled system, incoming traffic streams defined by different distributions converge to a Poisson distribution as time evolves. 'BP' refers to the service time distribution which is of Bounded Pareto (BP) type with service offered by a processor.

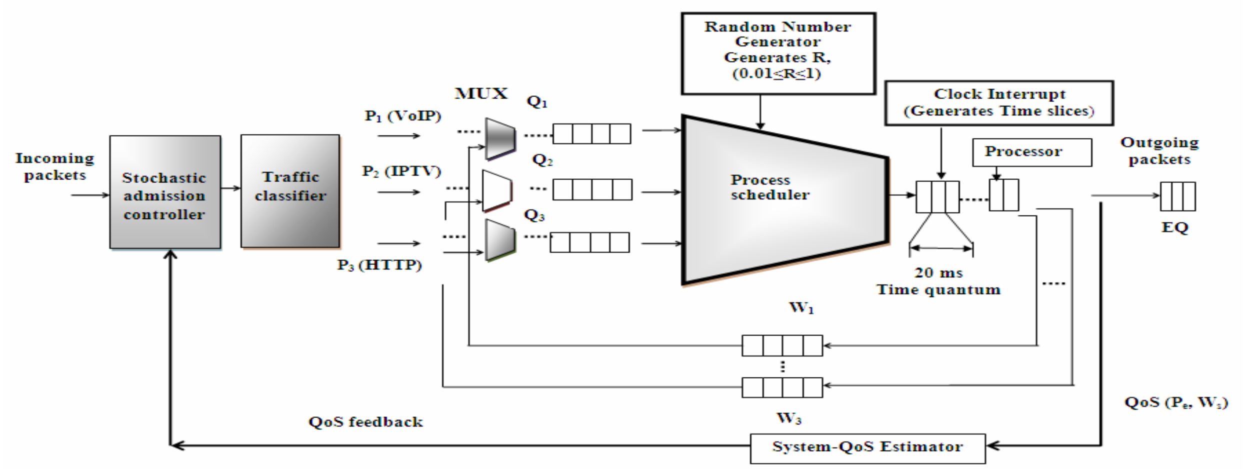

Figure 1. Illustration of M/BP/1/./QUEST model. Q_i: Ready queues, W_i: Waiting queues, EQ: Expired queue

The incoming processes are being scheduled according to QUEST scheduler. Service of each traffic is related with the defined value of QoS Class Identifiers (QCI). A lower value QCI denotes more restrictive services in terms of performance. The deadlines for VoIP, IPTV, HTTP are set considering acceptable practical deadline as stated in Table 1.

The priorities assigned to processes are inversely proportional to their deadline. Therefore, the priority of execution of processes are kept in the order of, $P_1 > P_2 > P_3$and process utilization ratio is provisioned as [0.8:0.16:0.04]. A clock interrupt generates the timing slices or *quanta*. After each slice, the next process is picked up from the ready queue. The scheduler runs through the ready queue, selects a process from a queue of processes to execute depending on the outcome of a random number generator, runs through the time slice, eventually placing the finished process in an expired queue. For practical real-time tasks, deadlines are in the range of 10-300 ms. Considering uniform burst time which is made possible by traffic conditioning algorithms like token bucket, leaky bucket, etc. process utilization (U_i) [14] of the system is expressed in Eq. (4).

$$\sum_{1}^{3} U_i = T_B \cdot \left(\frac{1}{D_1} + \frac{1}{D_2} + \frac{1}{D_3} \right) \le 1 \quad \textbf{(4)}$$

In this scheme, $T_B$denotes the burst time (service time) and the deadlines of processes are denoted by D_i. In case, $D_1=20$ *ms*, $D_2=100$ *ms*, $D_3=400$ *ms* the value of burst time is calculated as, $T_B \le 16$ *ms*. Allowing *4 ms* timing jitter (T_J) provides the required value of time quantum (T_Q). Thus, $T_Q = T_B + T_J = 20$ *ms*. The time quantum, T_Q, is set at 20 ms so that pre-emption does not result in deadline misses. This value of time quantum 20 ms is acceptable because it is at least equal to the minimum process deadline 20 ms, which is required for highest priority VoIP (process P_1) traffic to avoid context switching. Thus, designing the value of burst time as 16 ms justifies its use to keep the system utilization close to 100 percent.

4.1 QUEST scheduling algorithm

Algorithm clearly indicates that QUEST is a dynamic-priority scheduler because the next process to be executed depends purely on the outcome of the random number generator decided at run-time and may not have the highest priority among the pending processes.

```
Generate random number R, $0.01 \le R \le 1$;
Set: Time quantum $T_Q$: 20 ms and $T_B$ : 16 ms;
                              where $T_Q=(T_B+T_J)$
Set: Timing jitter (scheduling jitter): $T_J$;
    where, $0 \le T_J \le 4$ ms;
Initialize: timer, t=0
for t=1,2... ($T_B+T_J$) ms, do
switch (initial process) {
    CASE initial_process: $P_1$
        if ($0.01 \le R \le 0.9$) then
            execute $P_1$;
        else if ($0.91 \le R \le 0.98$) then
            execute $P_2$;
        elseexecute $P_3$;
        end if;
    CASE initial_process: $P_2$
        if ($0.01 \le R \le 0.56$) then
            execute $P_2$;
        else if ($0.57 \le R \le 0.95$) then
            execute $P_1$;
        else execute $P_3$;
        end if;
    CASE initial_process: $P_3$
        if ($0.01 \le R \le 0.40$) then
            execute $P_3$;
        else if ($0.41 \le R \le 0.82$) then
            execute $P_1$;
        else execute $P_2$;
        end if;    }
end for;
Place $P_i$ in expired queue;
```

4.2 Packet Loss Rate (PLR)

Among three multimedia traffic the deadline sensitive VoIP traffic shows highest PLR. According to G.711, G.729 and other compressing codecs VoIP must have far less than 1% PLR threshold.

In this work, the PLR is expressed as the root mean square error, $P_{e,rms}$, of L1, L2 cache miss and deadline miss errors of the system. $P_{e,rms}$ is stated in Eq. (9). L1 cache miss error, L2 cache miss errors and the deadline miss errors are denoted by C_{L1}, $C_{L2}$and D_e respectively.

$$P_{e,rms} = \sqrt{C_{L1}^2 + C_{L2}^2 + D_e^2} \qquad \textbf{(5)}$$

For each of the three processes: VoIP, IPTV, HTTP, the above r.m.s error is calculated from Eq. (5) and substituted in the second row of error probability matrix, *E*, given in Eq. (6).

5 Simulation Methodology

For simulation, an initial model is characterized by two matrixes, i) the TPM, *'T'* stated in Eq. (3) for the Markov model considered (here three-state model) and ii) *'E'*, an error (vector) probability matrix in (10). Practical values of cache miss errors and deadline miss error rates have been taken.

$$E = \begin{pmatrix} 0.98 & 0.9 & 0.8 \\ 0.02 & 0.1 & 0.2 \end{pmatrix} \qquad \textbf{(6)}$$

The three elements in the second row in Eq. (6) represent error probabilities of the processes and the elements in first row indicate the probabilities of correctness. The simulation framework has been developed using a discrete event simulator, DEVS suite [11] andMATLAB 2011R. Monte Carlo method has been applied for confirmation. Following (Table 2) system environment for simulation was used:

Table 2. Simulation parameters

Parameter	Conditions
Arrival rate	50 packets/s
Data file size	20-400 KB
Burst time	16 ms
Shape parameter (θ)	0.14
Service discipline	QUEST
Link Capacity (L_c)	10 Mbps
Packet size	1 KB
Simulation time	289 s
T_J	0.05 ms$\le T_J \le$0.5 ms

6 Simulation Result

6.1 Mean Waiting Time

In this subsection a comparative performance analysis in terms of mean waiting time, for QUEST, with current state-of-the-art scheduling algorithms-deferred pre-emption (DP), earliest deadline first (EDF)and accuracy-aware EDF (A-EDF) has been illustrated in Figure 2.

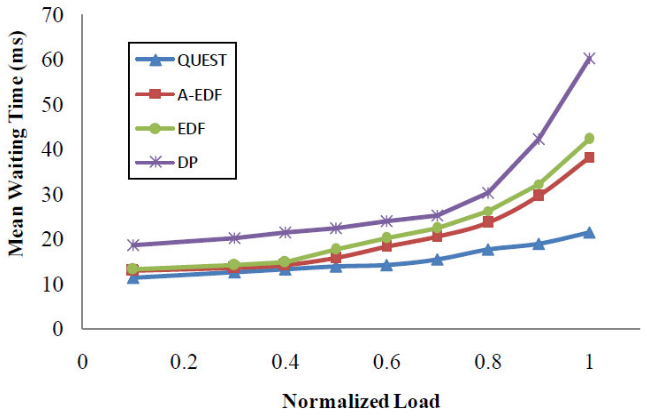

Figure 2. Mean waiting time with increasing load

Figure 2 shows that, QUEST experiences significantly lowest value of mean waiting timewith higher normalized load and it exhibits 23 percent improvement with respect to best competing A-EDF. Usage of a stochastic admission controller [12] which is permissible in QUEST, keeps the mean waiting time low even at high traffic loads close to 100 percent. EDF and its variant A-EDF are not stochastic, avoiding usage of such admission controllers. Therefore, for EDF, mean waiting time can be low only for loads below about 80 percent, which contradicts our original problem objective of close to 100 percent utilization. If stochastic admission controller is not used, in high load condition, the mean waiting time rise-rate would be much steep as happens with EDF and A-EDF depicted in Figure 3. Furthermore, RM (rate monotonic) as well as DP (Figure 3) are static priority scheduling algorithms and therefore, experience significant rise of mean waiting time with increasing normalized traffic load.

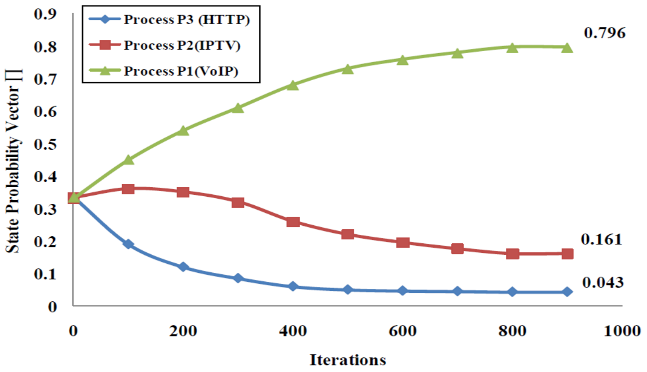

Figure 3. Convergence of State Probability Vector Π

6.2 Steady State Probability Analysis and System Stability

Simulation is performed considering random arrival of processes with the given error vector. The error vector provides error positions in 2000 sequences (iterations). The probability of finding the processor in a given state is calculated from *‘T’* and the error probability is obtained from *‘E’*.

As shown in Figure 3, Process P_1 (VoIP), Process P_2 (IPTV), and Process P_3 (HTTP) achieve steady state probabilities of 0.796, 0.161 and 0.043, respectively and the PLR (denoted as P_e)comes as 0.0045 (Figure 4), which is quite acceptable because it falls within the standard PLR threshold of 1 percent.

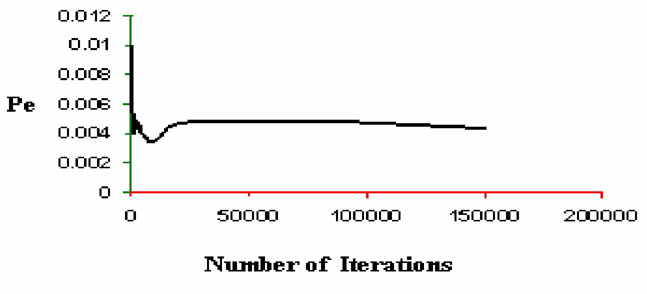

Figure 4. P_e converges to a steady state with number of increasing iterations

Thus, the lowest priority process traffic HTTP secures a guaranteed 4.3 percent process utilization. It justifies that low-priority process starvation is eliminated. Simulation is performed to calculate the packet loss rate (PLR) which is denoted as P_e. Results show that with the increasing count of sequences (iterations), P_e settles to a steady state value (shown in Figure 4).

7 Dynamicglobal Optimization and Re-configurability

PLR is to be minimized to optimize system performance. Due to the varying nature of load, the pre-allocated state transition probabilities of matrix *‘T’* are unfit to provision the QoS at its maximum. This problem is *solved* in a unique way byre-configuring the matrix *‘T’* using reconfiguration (tuning) parameters, Δ_1, Δ_2 and Δ_3 as stated in Eq. (11).

$$T_{recon} = \begin{pmatrix} 0.90-2\Delta_1 & 0.08+\Delta_1 & 0.02+\Delta_1 \\ 0.39+\Delta_2 & 0.56-2\Delta_2 & 0.05+\Delta_2 \\ 0.42+\Delta_3 & 0.18+\Delta_3 & 0.40-2\Delta_3 \end{pmatrix} \quad \textbf{(7)}$$

In this TPM, the diagonal elements are most dominant for a faster convergence. Therefore, for re-configurability (-2×Δ) has been subtracted, and Δs has been added with other two elements in each row in the TPM to make the total change to be zero. The sum of all the elements in a row of the TPM is ‘1’. These reconfiguration parameters drive the PLR to a *minimum* value and this will cause QoS back to maximum value by the feedback controller shown in Figure 5.

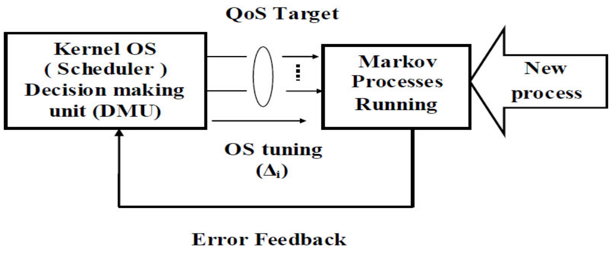

Figure 5. Feedback control system for re-configuring the QUEST scheduler

The system QoS is dynamically monitored by the scheduler using a feedback controller with the help of decision making unit (DMU) and necessary corrective actions are implemented.

The error feedback controller is used to reconfigure the QUEST by suitably tuning Δ_is. The 3D-contour plot of PLR (denoted as P_e) as function of Δ_1 and Δ_2 with $\Delta_3 = 0$) is shown in Figure 6.

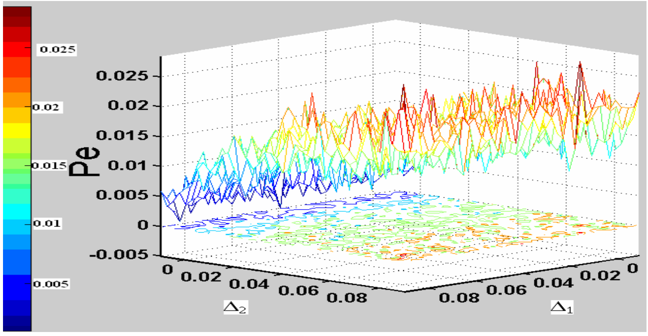

Figure 6. Re-configuration space of P_e vs. $\Delta_1, \Delta_2; \Delta_3 = 0$

Similarly, P_e can be plotted as function of Δ_2, Δ_3 and Δ_1, Δ_3. It has been noted that P_e is *globally minimum* at 0.001 if values of Δ_1, Δ_2, Δ_3 are kept at 0.025, -0.09 and 0, respectively-0.09 and 0, respectively.

8 Run-time Estimation of TPM by Machine Learning

Because the QUEST scheduling mechanism is *re-configurable* in nature, specific values of TPM parameters at a given time during system operation are *uncertain*. Therefore, it is essential to dynamically *estimate* the TPM parameters (elements of the matrix ‘T’) during operation. The transition probability matrix (TPM) parameters are estimated by a forward-backward *machine-learning algorithm* which learns during run-time from the observed error patterns (sequences) that serve as *training* data.

8.1 Stability and Accuracy of Run-Time TPM Estimation

As the process load varies on a demand basis within the system, the PLR changes accordingly. Therefore, the elements of ‘E’, the error probability matrix too changes with respect to time and iterations. The system simulates the newly estimated model having modified TPM. In this learning, Forward Backward algorithm is guaranteed to converge to a maximum log likelihood ratio (Figure 7)

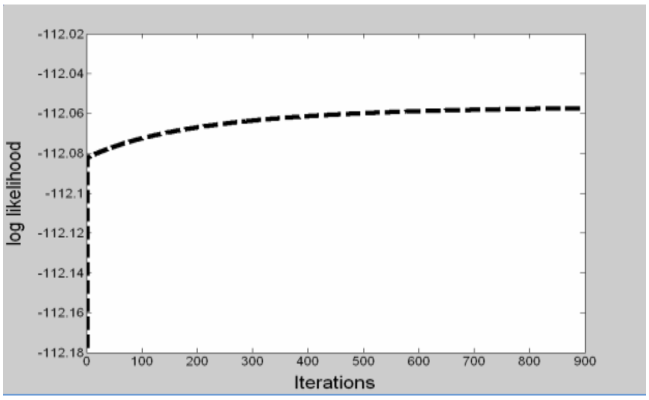

Figure7. Plot of log likelihood with respect to no. of Iterations

The convergence signifies stability of the system. The accuracy of the proposed scheduler is validated by comparing the run-time error patterns for initially considered TPM and for the estimated regenerated one. These patterns are illustrated in Figure 8.

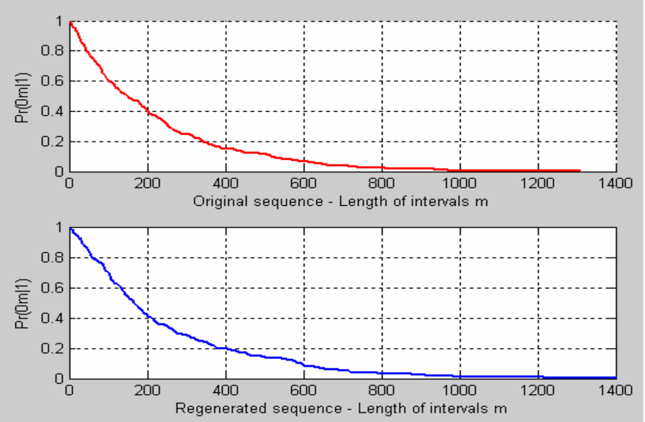

Figure 8. Pr $(0^m|1)$ for initial model and for newly estimated (regenerated) model

The two run-time error patterns are almost *identical*, confirming accuracy of the proposed model.

9 Performance Analysis of QUEST

9.1 Impact of PLR with Increasing Load

A comparative performance analysis of PLR (here, denoted as P_e) for current state of-the-art scheduling algorithms - earliest deadline first (EDF), deferred preemption (DP), accuracy-aware EDF (A-EDF) respect to QUEST for increasing normalized loadsare illustrated in Figure 9.

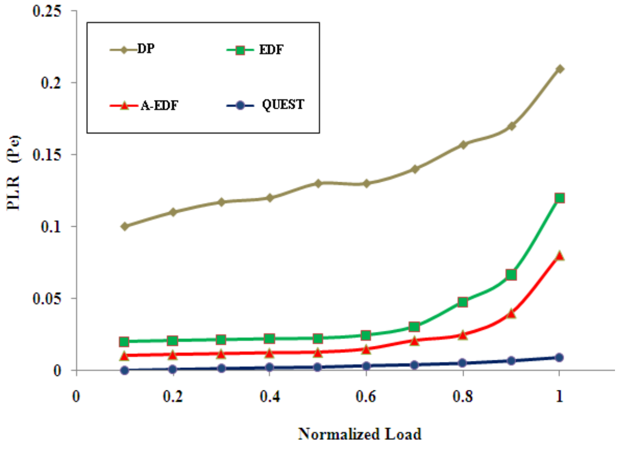

Figure 9. PLR for DP, EDF, A-EDF, QUEST

The L1, L2 cache miss errors and deadline miss errors for aforementioned schedulingalgorithms with typical values of L1=32 KBytes and L2=256 KBytes at a normalizedload of 0.9 are state in Table 3.

Table 3. Cache and deadline miss errors

Scheduler	L1 Cache miss	L2 Cache miss	Deadline miss
DP	0.051	0.119	0.068
EDF	0.0201	0.0469	0.0268
A-EDF	0.012	0.028	0.016
QUEST	0.0021	0.005	0.0028

It is observed from Figure 9 and Table 3, that QUEST scheduler outperforms other scheduling schemes and offers lowest value of PLR. The PLR is reduced by 37 percent in QUEST compared to A-EDF with lower values of cache and deadline misses. For QUEST, the improvement is due to use of Hidden Markov Model (HMM) filter (Baum-welch based) which is a probabilistic model applicable for finite and discrete process states. In contrast, A-EDF uses Kalman filter for process state estimation. Kalman filter is a special case of HMM applicable only for continuous and infinite states for a linear state space model which is not valid in digital embedded systems. Further, Kalman filter assumes Gaussian noise, whereas HMM filter makes no such assumptions and is thus more general and accurate. Furthermore, EDF and A-EDF have no explicit control on utilization, leading to unacceptably high deadline missratesat heavy loads. In stark contrast, QUEST enforces utilization close to 100 percent, making lower deadline misses even at heavy loads.

9.2 Impact of Scheduling Jitter and Scheduler's Noise Response

Considering the above scheduling framework (section 4), we investigate how the scheduling jitter affects the traffic flows and causes packet loss (PLR) during run-time. In order to evaluate the relation between jitter and PLR we perform stochastic simulations. Internet traffic is stochastic in nature. The scheduling jitter is defined as the unwanted variation of release times of stochastic processes by a scheduler within a router. The acceptable practical jitter for VoIP, IPTV and HTTP are illustrated in Table 4.

Table 4. Service model parameters for scheduling jitter

Traffic class	Acceptable Jitter (ms)	Recommended by
P_1 (VoIP)	≤ 0.5	ITU, network vendors
P_2 (IPTV)	< 30	network vendors
P_3 (HTTP)	400	network vendors

As it can be seen from Table 1 and Table 4, that VoIP traffic is most deadline-sensitive compared to other traffic flows with least value of acceptable jitter in the range of ($\leq$ 0.5ms). For VoIP any deviation from this threshold value of scheduling jitter would cause high level user dissatisfaction, violate service level agreements and more packet loss. As proposed, the QUEST scheduling algorithm (Sec. 4.1), we set the acceptable jitter (T_J) in the range of, $0.05 \leq T_J \leq 0.5$ ms and observe the PLR for each case. The increase of scheduling jitter will cause rise of PLR for each class which is plotted in Figure 10. The figure indicates VoIP experiences highest PLR.

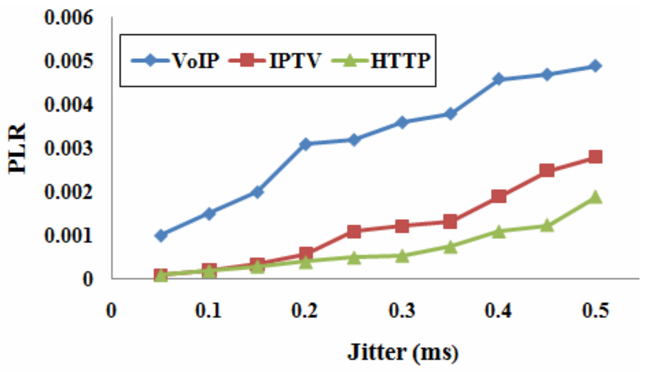

Figure 10. PLR for VoIP, IPTV and HTTP with increasing jitter

The resultant timing signal to noise ratio (SNR) is depicted in Figure 11.

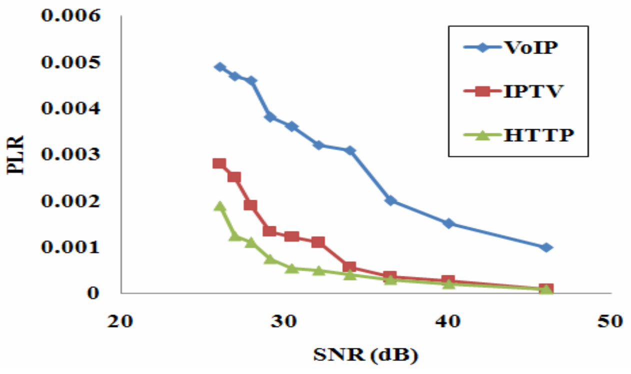

Figure 11. Plot of SNR vs. PLR

The graph shows that with increasing value of SNR, the PLR decreases.

10 Target Implementation Platform

As Intel's Atom processor is widely used in present day miniature embedded deviceslike routers, the proposed QUEST scheduler is here proposed to be implemented for Intel®'s Atom Z530 network embedded processor paired with Intel®'s System Controller Hub (SCH)US15W [13] to provide a platform for three different traffic processes shown in Figure 12.

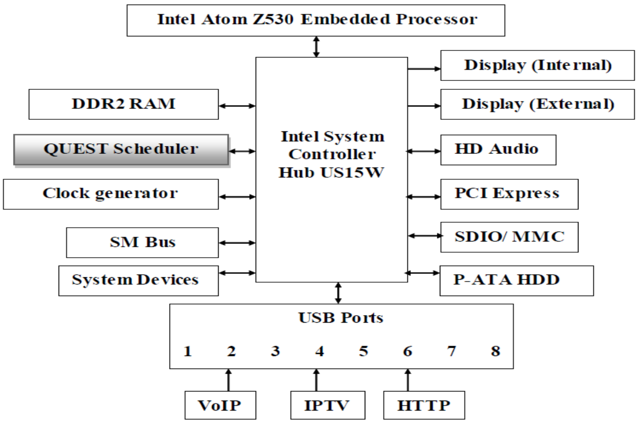

Figure 12. Proposed application of QUEST scheduler in Intel's Atom Z530 embedded processor with SCH US15W chipset platform

SCH is a single-chip component that contains the processor interface, DDR2 SDRAM controller, Intel Graphics Media Accelerator, various display interfaces, USB, SDIO, PCI Express, PATA, LPC, and other I/O capabilities. The three traffic streams enter the system through three USB ports having data transfer speed up to 480 MB/s. The scheduler is proposed to be implemented in the system management controllerdenoted as QUEST scheduler (in the figure), handles reset sequences, sleep state transitions for three processes, and other system management tasks.

11 Conclusion

This paper presents a novel re-configurable QoS-enhanced intelligent real-time stochastic packet scheduler - QUEST, for smart routers. Machine learning algorithms were used for dynamically optimize the system QoS during run-time and a utilization very close to 100 percent has been enforced. The scheduler's advantages are, solving the problem of priority starvation, arbitrary pre-programming of process utilization ratio and it addresses poor performance of the EDF scheduler at heavy loads. Performance of the scheduler was analyzed using QoS's important metrics: scheduling jitter, noise, PLR and mean waiting time. Simulation results indicate that the proposed scheduler outperforms current state-of-the-art schedulers.

References

[1] S. Li, J. Wen, F. Luo, M. Gao, J. Zeng, Z. Y. Dong, A New QoS-aware Web Service Recommendation System Based on Contextual Feature Recognition at Service-side, *IEEE Transactions on Network and Service Management*, Vol. 14, No. 2, pp. 332-342, June, 2017.

[2] B. Akbari, H. R. Rabiee, M. Ghanbari, Performance Analysis of Packet Loss Recovery Policies in P2P Video Streaming, *International Journal of Internet Protocol Technology*, Vol. 8, No. 1, pp. 44-53, 2014.

[3] M. F. Rabbi, C. C. Ko, Timing Jitter Tracking for Orthogonal Frequency Division Multiple Access System in High Doppler Spread, *IET Communications*, Vol. 6, No. 11, pp. 1438-1446, July, 2012.

[4] S. Jadhav, H. Zhang, Z. Huang, Quality of VoIP in WiMAX and UMTS, *Journal of Internet Technology*, Vol. 13, No. 3, pp. 491-499, May, 2012.

[5] C.-Y. Lin, L.-C. Yeh, C.-S. Wang, J.-W. Lin, Realization of Communications-as-a-service: IPTV Distribution Platform and a Network-based IPTV Audience Measurement Scheme, *Journal of Internet Technology*, Vol. 15, No. 4, pp. 565-575, July, 2014.

[6] QoS information – Knowledgebase of OTEL Communications, Practical Values of Acceptable Jitter for VoIP Traffic Specified by Various Vendors, http://billing.otelafrica.com/knowledgebase. php?action=displayarticle&id=125.

[7] M. Short, Bounds on Worst-case Deadline Failure Probabilities in Controller Area Networks, *Journalof Computer Networks and Communications*, Vol. 2016, pp. 1-12, Article ID: 5196092.

[8] H.-G. Kim, Enhanced Timing Recovery Using Active Jitter Estimation for Voice-over IP Networks, *KSII Transactions on Internet and Information Systems*, Vol. 6, No. 4, pp. 1006-1025, April, 2012.

[9] C. Ghazel, L. Saïdane, Satisfying QoS Requirements in NGN Networks Using a Dynamic Adaptive Queuing Delay Controlmethod, *Procedia Computer Science*, Vol. 56, pp. 225-232, December, 2015.

[10] R.-D. Huang, S.-Y. Kuo, Y.-H. Chou, Detecting Strategy of Fast Flux Domain Based on Hidden Markov Model, *Journal of InternetTechnology*, Vol. 16, No. 2, pp. 277-287, March, 2015.

[11] Arizona Center of Integrative Modeling and Simulation of Arizona State University, *DEVS Suite: Discrete Event System Simulator Suite*, http://acims.asu.edu/software/devs-suite.

[12] A. Peculea, B. Iancu, V. Dadarlat, I. Ignat, A Novel QoS Framework Based on Admission Control Andself-adaptive Bandwidth Reconfiguration, *International Journal of Computers, Communications & Control* (Journal of Agora University), Vol. 5, No. 5, pp. 862-870, December, 2010.

[13] Intel® System Controller Hub (Intel® SCH) Datasheet, https://www.intel.com/content/www/us/en/intelligent-systems/menlow/sch-datasheet.html

Biographies

Suman Paul is currently an Assistant Professor in the Department of Electronics and Communication Engineering, Haldia Institute of Technology, West Bengal University of Technology (Maulana Abul Kalam AzadUniversity of TechnologyWest Bengal), India. He interests in scheduling and QoS in telecommunication networks.

Malay Kumar Pandit is currently a Professor in the Electronics and Communication Engg Dept., Haldia Institute of Technology, India. Dr. Pandit received his Ph.D. degree from Cambridge University, UK and post-doc from the City University of Hong Kong. He interests in scheduling and QoS issues in communication networks.