Implementation of Class-Based Low Latency Fair Queueing (CBLLFQ) Packet Scheduling Algorithm for HSDPA Core Network

Sohail Ahmed[1*], Malik Muhammad Asim[2], Nadeem Qaisar Mehmood[3], Mubashir Ali[4], and Babar Shahzaad[5]
[1,3,4]Department of Computer Science, University of Lahore
Gujrat, 50700 - Pakistan
[e-mail:[sohail.ahmed],[nadeem.mehmood],[mubashir.ali]@cs.uol.edu.pk]
[2]Department of Computer Engineering, University of Engineering and Technology
Taxila, 47080-Pakistan
[e-mail: asim.malik@uettaxila.edu.pk]
[5]Department of Computer Science, University of Management and Technology
Sialkot, 51040 – Pakistan
[e-mail:babarshahzaad@gmail.com]
*Corresponding author: Sohail Ahmed

Received September 19, 2018; revised November 17, 2018; revised August 10, 2019; accepted October 10, 2019; published February 29, 2020

Abstract

To provide a guaranteed Quality of Service (QoS) to real-time traffic in High-Speed Downlink Packet Access (HSDPA) core network, we proposed an enhanced mechanism. For an enhanced QoS, a Class-Based Low Latency Fair Queueing (CBLLFQ) packet scheduling algorithm is introduced in this work. Packet classification, metering, queuing, and scheduling using differentiated services (DiffServ) environment was the points in focus. To classify different types of real-time voice and multimedia traffic, the QoS provisioning mechanisms use different DiffServ code points (DSCP).The proposed algorithm is based on traffic classes which efficiently require the guarantee of services and specified level of fairness. In CBLLFQ, a mapping criterion and an efficient queuing mechanism for voice, video and other traffic in separate queues are used. It is proved, that the algorithm enhances the throughput and fairness along with a reduction in the delay and packet loss factors for smooth and worst traffic conditions. The results calculated through simulation show that the proposed calculations meet the QoS prerequisites efficiently.

Keywords: Packet Classification, Marking, Scheduling, QoS and DiffServ.

http://doi.org/10.3837/tiis.2020.02.001 **ISSN : 1976-7277**

1. Introduction

In the modern era, the telecommunication and internet work together to fulfill computer programming and engineering challenges. The transmission of voice as a packet over an IP network, that is known as Voice over Internet Protocol (VoIP), is a basic requirement of the modern age. The people have adopted VoIP technology due to its low cost and free calling services. This popularity demands the Quality of Service (QoS) for real-time voice and video services. The internet doesn't guarantee QoS during data communication [1], along side the unfairness in traffic handling is another challenge for QoS provisioning in the communication network. The DiffServ network is affected by these problems [2,3]. Our focus, in this research, is to improve and ensure the QoS and fairness in real-time voice and video traffic. A mechanism is required through which the available bandwidth is efficiently managed between different types of traffic. An advanced wireless network can handle different real-time traffics like video, VoIP and other non-real time traffic applications [15].

The HSDPA network should have the ability to handle real-time and non-real time traffic with a guarantee of QoS [4]. The Integrated Services (InterServ) [22] and Differentiated Services (DiffServ) are the two different mechanism standardized by IETF,to provide a QoS in current IP networks [20]. But InterServ has complexity and scalability issues [22], while DiffServ can be implemented for HSDPA using a simple mechanism. In DiffServ the code points,also known as Differentiated Services Code Points (DSCP), are attached within the header of a packet. Different DSCP values are used by network boundary routers, having Per Hop Behaviour (PHB), to classify the traffic into different categories [5]. The lack of a standard implementation of PHB has gained noteworthy attention of the researchers to fill this gap.

To gain an end to end QoS provisioning, a layered architecture is defined by 3GPP standards [17]; where a Bearer Service (BS) assignment functionality is implemented from the perspective of the source to the destination, and this has all parameters to facilitate service provisioning functions for the QoS [8]. The BS of the universal mobile telecommunications system (UMTS) behaves according to the QoS profile to provide a desired level of service to a user. The four QoS service specifications defined by UMTS are: i) streaming, ii) conversational, iii) interactive and iv) background [6]. These classes are categorized according to their delay sensitivity factor. The streaming and conversational classes are for the real-time applications as they are more delay sensitive [14]. Whereas, the interactive and background classes are for soft or non real-time applications as they are less delay sensitive, therefore delay specifications are not required while transmitting packets. To decrease the delay in the real-time traffic, a mapping of these UMTS services into DiffServ classes is required as follows.

The mapping of DSCP is done in the edge router according to the delay sensitive classes[12]. DiffServ have three main classes as: i) Expedited Forwarding (EF) PHB, dedicated to low-loss, low-latency traffic and low jitter service [25, 26]; ii) Assured Forwarding (AF) PHB, gives assurance of delivery under prescribed conditions; and iii) Best Effort (BE) PHBs, which maintain backward compatibility with the IP precedence field [20]. In EF PHB class, the traffic is treated with high priority. The conversational class traffic i.e., VoIP is comparatively very much sensitive to packet delay; so is suitable for EF DiffServ class. The Assured Forwarding per-hop behavior (AF PHB) class , specified by IETF,is suitable for streaming service [19]. The proposed scheduling algorithm works on different classes of traffic and provides guaranteed QoS to real-time voice traffic in HSDPA core network. For the provisioning of QoS in a UMTS network, a project of SEACORN [16, 23] has contributed towards the implementation and development of a Radio Resource Management (RRM) framework. The Enhanced UMTS Radio Access Network-Extension (EURANE) is an extension of UMTS in NS-2 (Network Simulator), which contributed to the SEACORN project. We opted this EURANE tool to simulate the UMTS QoS scenario.

2. Literature Review

The architecture and concept of QoS for HSDPA network, defined by 3GPP, deals with the QoS of signaling between boundary routers and user entity [18, 27]. This architecture is unable to handle the procedure of QoS for the transportation of data packets. For an optimized provisioning of end-to-end QoS in HSDPA, we should find better mechanisms to ensure QoS for user data transportation and mapping of IP network class into UMTS network class [6]. Following sequence of research works present previous similar mechanisms:

In [15], the authors focused on customized mapping of video and voice real-time traffic to guarantee an improved end-to-end QoS in HSDPA network. However the non real-time traffic, belongs to interactive and background classes, also needs attention with improved mechanisms to guarantee a desired QoS.

The authors,in [24], used Weighted Round Robin (WRR) to reduce delay in packet throughput; but this generated traffic fairness issue as it did not treat traffic classification equally, which is an important factor. WRR is also affected by the variable size of the packets in the queue.

The authors used Deficit Round Robin (DRR) as Low Latency Queuing (LLQ) to provide fairness to all traffic's flow [9]. The jitter and fairness issues are efficiently handled by DRR in both high and slow speed networks. Simple and fair packet scheduling make able the DRR to provide a guaranteed service. But if the packet weight and size vary, the flow'sjitter and the delay is increased. Calculation of each packet size also affects the efficiency of DRR scheduler.

On the other hand, there are many schedulers which are not affected by this problem; so provide efficient and fair scheduling [11]. These schedulers guarantee a near optimal deviation according to anyweight and packet size distribution of traffic flow. Quick Fair Queuing (QFQ) is the fastest of all, but is two times slower than the DRR, since this increases the packet delay.

To ensure QoS for real-time voice traffic, the Weighted Fair Queuing (WFQ) scheduling algorithm was used with Priority Queues (PQ) [10]. The complexity level of the algorithm was increased because it calculates the start and end time of each packet, during the weight assignment to the packets. This was the cause of increase in delay in voice and video traffic delivery [28]. It also classifies packets according to their service category but assigns equal network bandwidth to each class, so there is fairness issue too.

The rest of the paper is organized as: Section 3 discusses the important functions required for packet scheduling . Section 4 describes the proposed scheduler and its necessary factors. The guarantee of service factoris also elaborated in this Section. The simulation scenario is discussed in Section 5. Section 6 illustrates the experimental results and the application perspective of proposed research is discussed in section 7.We conclude the research work in Section 8.

3. Preliminaries and Related Algorithms

Here, we discuss the background factors and related algorithms.

3.1 Background Factors

The classification and marking of traffic are performed using Improved Time Sliding Window Three Colour Marker (ItswTCM) [21]. There are two basic components of ItswTCM, the rate estimator and the marker. The rate estimator is used to measure the average running bandwidth that helps to handle the case of busty and normal traffic; whereas the marker provides probabilistically mechanism to assign a color based on estimated traffic rate. The colors represent the drop precedence level in Assured Forwarding Per-Hop Behavior(AF PHB). After marking the packet, the color is translated into DSCP representing 0, 1 and 2 levels of precedence in a single AF class using descending order.

3.2 Metering Function

In ItswTCM metering function is performed by rate estimator, with its basic purpose to calculate the average arrival rate of the traffic stream [1].After calculating the average rate, the ItswTCM performs the marking operation. For this purpose, the average running bandwidth, represented by `Time_Avg,` for a specific time period is calculated. ItswTCM calculates the transmission rate based on arrival packet and decay to maintain the state of three quantities: i) `Avg_Rate,`to estimate the packet arrival rate; ii) `Time_Avg`, known as a window, representing a specific time unit; and iii) the `Last_Time`, representing the arrival time of the last packet. The values of `Avg_Rate` and `Last_Time` variables are updated on the arrival of each packet, but the `Time_Avg` or window length is static and predefined. ItswTCM remembers the `Time_Avg` using a maintained history by estimating the rate of sent packets. The `Current_Time` is the arrival time of the current packet and its packet size is represented by `Pkt_Size.`We measure the traffic rate and assign a color to packets depending on two basic type of defined rates i.e., Committed Target Rate(CTR) and Peak Target Rate(PTR) [7]. Initially, the CTR is our `Avg_Rate` and the `Last_Time` is zero.

The algorithm receives the ‘N’ packets as an input. For each packet it first calculates the `Bytes_in_interval` with the help of `Avg_Rate` and `Time_Avg` or window size. To calculate the space in the window for the new arrival bytes (i.e. New_Bytes), it uses the calcualted `Bytes_in_interval` and the size of the currently arrival packet i.e. `Pkt_Size.`Then the average rate will be estimated by dividing the newly arrival bytes (i.e. `New_Bytes`) with the submission of the time difference of packet arrival and the window size (i.e. `Time_Avg`). The algorithm then updates the `Last_Time` state variable with the arrival time of the currently arrived packet. This estimated averagerate is then assigned to the marking function where suitable color will be marked.The metering function as illustrated in **Function 1**.

```
Initial value\\
Last_time =0
Avg_Rate = CTR
While(N)
Bytes_in_interval = Avg_Rate * Time_Avg;
New_Bytes = Bytes_in_interval + Pkt_Size;
Avg_Rate = $\frac{New\_bytes}{(CURRENT\_TIME - Last\_time + Time\_Avg)}$;
Last_Time = Current_Time;
Marking_function(Avg_Rate)
```

Function 1. Metering Function

3.3 Marking

To achieve customized services, we should assign some priority to the desired traffic [7]. The tswTCM algorithm suffers from traffic fairness issue[11, 2]. To transmit more yellow packets, ItswTCM marking mechanism uses a constant *C* with CIR condition which is greater than one [1]. In this way it can handle unfair bandwidth sharing problem. If the throughput of traffic is less than or equal to the CTR then ItswTCM will assign the green color to the packets. If the packet's throughput lies between CTR and PTR, then the yellow color will be assigned; however if the packet's throughput exceeds the PTR, then the red color is assigned. Different types of traffic, traffic model, size of packets and transport protocols are shown in **Table 1**.

Table 1. DiffServ class mapping and traffic model

Traffic Type	**Application**	**Model for Holding Time**	**Traffic Model for Application**	**DiffServ DSCP**	**Size of IP Packet (byte)**	**Transport Protocol**	**DSCP Value**	**RFC**
Streaming	Multimedia Streaming	Exponential	On / Off EXP.	AF11	500	UDP	26	2597
Conversational	Voice	Exponential	On / Off EXP.	EF	250	UDP	46	3246
Background	FTP	Pareto	CBR model	BE	1000	TCP	0	2474
Interactive	Web	Log Normal	On/Off Pareto	AF21	1000	TCP	8	3662

3.3.1 Marking Algorithm

The marker's responsibility is to determine the color of the packet based on estimated rate during metering. A marker ensures that if the average estimated rate is less than or equal to CTR, then assign the green color to packets. If the average estimated rate is less than PTR but greater than CTR; then assign the yellow color to packets with P_0 probability and assign $(1 - P_0)$ probability to green color. P_0 is a value of packet contribution in calculated stream rate. This function is illustrated in **algorithm 1**.

```
RATE_AVG = Average rate of packet sending
if (RATE_AVG > PTR)
Find P1 = $\frac{RATE\_AVG - PTR}{RATE\_AVG}$

Find P2 =$\frac{PTR - CTR}{RATE\_AVG}$
    Assign a Red color to packet with P1 probability;
    Assign a Yellow color to packet with P2 probability;
    Assign a Green color to packet with (1- (P1+P2)) probability;

else if ((RATE_AVG > CTR) AND (RATE_AVG ≤ PTR))
    Find P0 = $\frac{RATE\_AVG - CTR}{RATE\_AVG}$
    Assign a Yellow color to packet with P0 probability;
    Assign a Green color to packet with (1-P2) probability;

else (RATE_AVG ≤ CTR);

Assign a Green color to the packet;
```

Algorithm 1. Marking Algorithm

3.4 Fairness Factor

The fair queuing mechanism is the basic need of scheduler. To ensure this, the scheduler uses a global variable known as *h(t)* virtual time. The variable's responsibility is to maintain a record of scheduler's work [29]. The virtual time is represented in equation 1 as:

$$\frac{dh(t)}{dt} = \frac{s}{\sum_{j\to f(t)} sj} \tag{1}$$

In equation 1, *s* is server rate, and *f(t)* represents different flows that are saved by the scheduler at time *t*. If apacket's arrival rate is high, then the *rising rate* of virtual time will increase. On the other hand, if the out flow of the packet departure becomes idle then the rising rate of virtual time will reduce. So, the arrival and departure of packets update the virtual time value. In the worst scenario, for *G* packets in *f(t)* flows, the updating value function manages *G* event(s). The system computes the time stamp as a component of *h(t)* to attach it with each packet and to deliver accordingly. For WFQ the scheduler performs these calculations against each packet, but in the case of Deficit Weighted Round Robin (DWRR), these calculations are performed for specific *queues*, instead of each *packet*.

3.5 Mapping of DiffServ into UMTS QoS Parameters

The DiffServ classes provide the guarantee to deliver packet according to a required rate as definedin the Service Level Agreement (SLA) for a particular user. During congestion on the link, the traffic crosses the threshold of the boundary rate, therefore creates high probability for the packets to get dropped [13]. There are four DiffServ classes, as previously mentioned, with three levels of dropping precedence each, which are implemented in "Assured

Forwarding PHB" to control the data packets [19]. Each class is specified with different dropping levels and buffering configuration for sharing bandwidth. The mapping of UMTS classes to DiffServ codes is shown in **Table 1**.

The streaming class is less sensitive to delay so is mapped with the AF PHB class. The interactive class has no limit condition for delay and reliability, so is mapped with Best Effort (BE) class that is also known as the default class. The marking function of DSCP, that are our policy, is activated at the edge router; which allocates the code points to each packet. According to the DSCP values, different IP packets are handled by different virtual/physical queues. The packet's bottleneck problem is handled through the buffering and policy algorithm, in the outgoing link of Gateway GPRS Support Node (GGSN) and Serving GPRS Support Node (SGSN). Our proposed algorithm improves the network line consumption during the bottleneck and reduces the blocking rate of a session. As a result, the overall throughput is improved. Once marking and queuing mechanism of the packets are performed, then in the scheduling phase the scheduler improves the efficiency by sharing common resources between different packet delivery serving classes. The task of the scheduling algorithm is to manage the available core network's resources between the authorized user groups. In case of link congestion the scheduling mechanism is required which handles the queue operations; whereas in case of no congestion, the scheduler transmits all the arrived packets without any delay.

4. Proposed CBLLFQ Scheduling Algorithm

Our proposed scheduling algorithm offers less delay, less packet loss and good throughput with fairness property. This algorithm also reduces the complexity level using DWRR. The real-time traffic, like VoIP, demands less packet loss and less delay, and our algorithm provides these QoS parameters. To solve the delay issue in real-time applications, a new CBLLFQ scheduling algorithmis introduced which uses a strict Priority Queue (PQ) integrated with DWRR. The priority or weight based distribution of network bandwidth helps to improve the fairness property in this algorithm. While using priorities within in queues it can avoid starvation, hence guarantied bandwidth is provived to non-real time traffic as well. This mechanism is also suitable for jitter and delay sensitive traffics. The steps of the proposed CBLLFQ scheduling algorithm are illustrated in **Fig. 1**.

The algorithm is a combination of two basic types of the scheduling algorithms, PQ and DWRR, to achieve best bandwidth utilization by keeping the loss and delay of IP data packets within the limits.There are four physical and three logical queues to ensure different levels of drop precedence. The major concern in this research is to provide priority to the real-time voice traffic. To achieve this goal, VoIP traffic was mapped with EF class using classification, marker operation and enqueuing operation in the first queue according to the PQ algorithm. Whenever there is no packet in the first queue, the other three queues will be treated according to the DWRR algorithm. During this process, if a packet comes into the first queue, the scheduler will save the quantum time of served queue in the log and move to the first queue for the real-time processing of the packet.

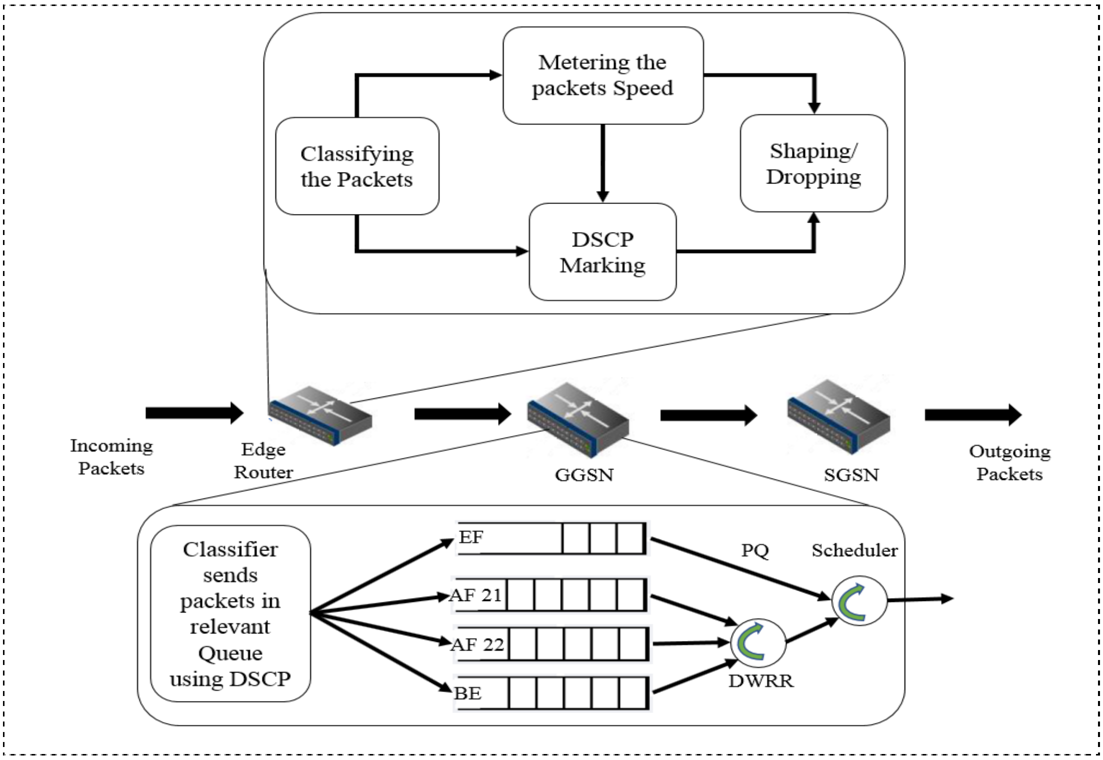

Fig. 1. Traffic Conditioning and Scheduling Operation in DiffServ Domain

After serving it, the scheduler will start serving another queue according to the saved logs. The second physical queue is reserved for video streaming, the third for web services, and the fourth for FTP service. So, after classification and marking,the video traffic, web traffic and FTP traffic will be mapped with AF21, AF22 and BE classes respectively. Hence after mapping, all the packets will be enqueued in their corresponding queues.

The CBLLFQ deals the three low priority physical queues with two basic entities i.e. the quantum size and the weight assigned. Quantum size is basically the window size measured in bytes; and that can be transmitted at a single unit of time. Whereas the weight is defined as the fairness level or priority level of the queue that can be achieved through CBLLFQ. If weight or priority level of one queue is higher than the other, then higher priority queue can send more numbers of quantum than other. We have defined weights for 2nd, 3rd and 4th queues as three, two and one respectively. If a packet comes in the priority queue it will be served first. After serving all the packets from that queue, the scheduler will serve remaining three queues using DWRR algorithm. Due to the limit of queue's size, a fixed number of packets are storeable. In an ideal condition, none of the packets may store in any queue during any time unit. This means that all arrived packets will departure without any delay, hence will result into the ideal QoS. But during burst condition, several number of packets will be stored in the physical queue. During this stage, CBLLFQ will play an important role to ensure user's Service Level Agreement (SLA). This will assign a high priority to the real-time SLA user and will give less priority to the non real-time SLA user.In worst condition the queues are filled with packets, so the next arriving packets will be dropped by Random Early Detection (RED) algorithmas shown in **Fig. 2** and **algorithm 2**. RED, a congestion avoidance mechanism, uses adrop tail method to drop the packets. Upon receiveing a packet RED compares the packet size with the queue size. If there is some empty space available for the new arrival packet then CBLLFQ scheduling algorithm stores it, otherwise will drop by RED. According to DWRR, the computed total number of bytes in a single block are in equation 2 as:

$$B[i] = N \times W_i \tag{2}$$

The B$[i]$ represents total bytes transmitted from queue i in a single time unit, where N is the quantum value and W_i is the weight of the i^{th} queue. The value N is constant for each queue but weight W_i is variable according to the priority levels. The number of dequeued bytes in a queue will vary depending on the weight. From equation 2, it is clear that the DWRR will dequeue a greater number of bytes from high priority queue as a comparison to the low priority queue. The dequeue function will be performed until each packet from each queue is dequeued. The classification, marking and scheduling functionsare illustrated in **Fig. 2** having **algorithm 2**.

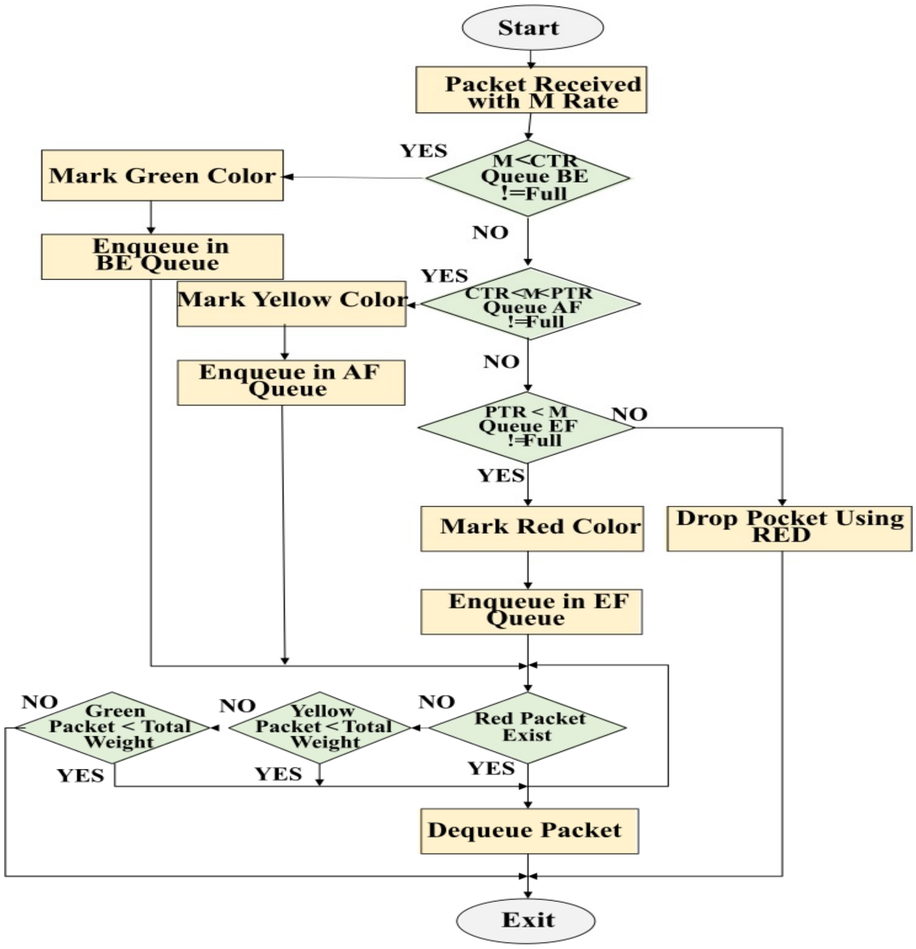

Fig. 2. Proposed classification, marking and scheduling algorithm

Case: If the total number of bytes in the i^{th} queue are less than B*[i]*, the log will update the total balance and remaining balance using the following equation 3.

$$R[i] = B[i] – S[i] \tag{3}$$

Where the remaining balance of bytes in the queue is R*[i]* and total served bytes are S*[i]*. In the next round, this remaining balance will be added to total balance B*[i]*. This rule is also applied to the other two queues. After the arrival of the packet, the scheduler checks its length ln_i and saves (i, ln_i) entity in the backlog of the relevant queue. The scheduler manages a record using some necessary parameters, which are: W*[i]* and Q*[i]* representing weights and sum of quantum values respectively. D_i represents the four physical queues for four types of traffics. The *t* and *(t-1)* represent current and previous time of the packet. Initially, the weight of queues, quantum values, length of packet and time are zero but will be updated by a scheduling function 1 and 2 during each iteration.

4.1 Scheduling Functions

At the initial stage, when the packet is received by the scheduler, it increases the quantum values Q*[i]* by weight of that queue W*i* and calculates packet's length *L*. The sum of remaining quantum for each queue is updated during each iteration. **Function 2** and **3** describe the basic functions of enqueue and dequeue for DWRR, as depicted below with comments.

```
Input: Packet Ni                     // From ith Flow
While (Ni)
$lni = lni + (lni / W[i])$           // Compute Packet Length
$Qi = Qi + W[i]$                     // Assign quantum
Queuej ← Enqueue Ni
$lni = \max(Qsize, lni) + (lni / W[i])$   // Computes the remaining serving time according to Length of
Queue size
$(i, lni) = (i-1, ln(i-1))$          // Save in backlog.
t = t-1                              // Update the time
```

Function 2. Enqueue function for DWRR

```
Input: Token i                       // For dequeue Ni from Queue j
While(Token i)
$Qi = Qi - W[i]$                     // Reduce the Quantum value according to queue's weight
SGSN ← Dequeue Ni from Queue j
t = t-1                              // Update the time
$Qsize = Qsize + Qi$                 // Update the queue size
$(i, lni) = (i-1, ln(i-1))$          // Save in backlog.
```

Function 3. Dequeue function for DWRR

Algorithm 2 uses these functions to perform enqueue and dequeue operation from DWRR reserved queues. The algorithm is briefly discussed in the proposed scheduling section.

```
//Enqueue Function
Input: Packet” N”
While(N)
If ((Packet Rate M < CTR) && (Total bytes G < Total size of BE))
N ← Mark Green color
BE Queue ← Enqueue N using Function 2
Elseif ((CTR < M < PTR) && (Total bytes H < Total size of AF))
N ← Mark Yellow color
AF Queue ← Enqueue N using Function 2
Elseif ((M > CTR) && (Total bytes I < Total size of EF))
N ← Mark red color
EF Queue ← Enqueue N using Function 2
Else   Drop packet using RED
//Dequeue Function
If (G! = Null)
SGSN←Dequeue N from EF Queue
Elseif ((Packet weight AF Queue < Defined Weight) && (H! = Null ))
SGSN←Dequeue N from AF Queue using Function 3
Elseif ((Packet weight BE Queue < Defined Weight) && (I! = Null ))
SGSN←Dequeue N from BE Queue using Function 3
```

Algorithm 2. Packets Enqueue and Dequeue Algorithm

4.2 Quantum Update

The traffic profile and load received by scheduler are checked and upgraded on each interval. The assigned weight and quantum values are affected by QoS parameters like throughput, packet delay and buffer capacity. In the proposed algorithm, the quantum value and weight are adjustable to ensure the best scheduler according to the QoS parameters of different traffic classes, in different burst and smooth traffic situations. The adjustment and updation of weight and quantum values are performed logarithmically using exponentially and normalized parameters on different queue sizes. Log section mitigates the fluctuations of the quantum value effect and exponential section increases the burst effect on weight. Basically, W*[i]* weight of different priority classes is defined higher than other classes' values. When the size of the queue is small as $Qsize < (Qmax / 4)$ then we allow very less traffic flexibility but; when the size of the queue is large as $Qsize > (Qmax / 4)$ then the weight is flexibily changeable however the index is extracted by a combination of quantum values. These changes are described as:

if Qsize < ($Qmax$ / 4)

$$W[v] = Gj[i] \times e^{stream[i]} + [W0[i] \times (1+\log(1+Q[v]/Q[total]))]$$

Else

$$W[v] = W0[i] \tag{4}$$

According to equation 4, the index of traffic increases in the burst case and for this the queue has to be managed accordingly. High and medium traffic states are dealt while depending upon the size of the queue. Regarding size of the queue, if $Qsize < (Qmax / 4)$ then this is a high traffic state; and if $(Qmax / 4) < Qsize < (3Qmax / 4)$ then this is a medium traffic state. The two vales of $G_j[i]$ are referred to handle the two high and medium levels of traffic states.

In the first case, if Q_{size} exceeds Q_{max} /4 then according to equation 5, we can change the weight of queue W*[v]* exponentially; and the quantum value Q*[v]* logarithmically, with the main goals to get efficient QoS for different traffics.

$$Gj[i]= \begin{cases} G1[i] \text{ if } (Qmax/4) < Qsize < (3Qmax/4) \\ G2[i] \text{ if } (3Qmax/4) < Qsize \end{cases} \quad (5)$$

The value of *Gj* in equation 5 is a coefficient of our exponential section that should satisfy below-mentioned properties:

- *If the high priority traffic increases then, the G_j value of low priority traffic should be decreased to get better resource distribution against each traffic class.*
- *The G_j value of high priority class should be higher than low priority class in case of an equal load of traffic.*

We can express G_j value for the total number of queues *i* according to the variation of traffic load ω. As the scheduler moves towards low priority queue the value of G_j should be decreased. This operation is presented in below expression where the value of *m* is considered in ascending order from high to low priority queues.

$$Gj[i] = ƛj(1- \sum_{m-1}^{n-1} \omega i) \text{ where j=1,2} \quad (6)$$

In equation 6, we calculate two values *G1[i]* and *G2[i]* for high and medium traffic. We also find *ƛj* from the impact of burst traffic with weight.

4.3 Guarantee of Service

To compute the guarantee of service provided by the proposed scheduling algorithm, we consider the fairness factor of a different class of traffic. We compute the minimum level of guarantee by CBLLFQ in terms of bits sending from the i^{th} number of flows. This transmission of bits should be considered within a specified time interval *[ts,te]*in which the quantum and weight index are updated continuously. To calculate the overall guarantee of service, we define the total weights of all queues by Wi*[ts,te]* in the i^{th} flow. We provide an opportunity of sending bits in an interval of *[ts,te]*also. This opportunity is represented as *z*, *ts* as starting time, and *te* as ending time of bit transmission for i^{th} flow.

$$W[ts,te]=[z \sum_{k=0}^{l} \frac{\omega k}{\omega min} M + (l-1)M] > [\sum_{k=1}^{l} W[ts,te]\ ]$$

$$W[ts,te] = \frac{z\sum_{k=1}^{l} \omega k}{\omega min} M + (l-1)M \quad (7)$$

In equation 7, ω^k represents the weight of the kth flow and ω_{min} represent the lowest weight of all flows. If the packet size of any flow is less than *M*, then the bound will become loose. We consider the mentioned loose bound to avoid much longer formulae. To get a minimum level of guarantee of service for the kth flow, the expression can be used as:

$$(z - 1) \frac{\omega k}{\omega min} M - M \leq W[ts,te] \quad (8)$$

In equation 8, the entity *(z-1)* is not equal to the *z* in equation 7, because there is a chance that *te* may become equal to the sending opportunity. In such a case the flow may not use the opportunity at time *te*. Now, to formulize the upper limit of weight *W[ts, te]* as a function of *Wi[ts, te* , we follow equation 8 for z as:

$$\frac{\omega min}{\omega i}\left(\frac{\mathrm{Wi}[ts,te]}{M} + \frac{\omega k}{\omega min} + 1\right) \geq \mathrm{Z} \tag{9}$$

To compute the limit of transmission of bits in a specific time period, in which *Wi[ts, te]* bits of the i^{th} flow are transmitted; we replaced the inequality of equation 9 in equation 7 as:

$$\mathrm{W}[ts,te] = \left[\frac{\sum_{k=1}^{l}\omega\mathrm{k}}{\omega i}\,\mathrm{M} + (l-1)\mathrm{M}\right]\left[\frac{\mathrm{Wi}[ts,te]}{M} + \frac{\omega k}{\omega min} + 1\right] \geq \mathrm{W}i[ts,te]$$

$$\mathrm{W}[ts,te] = \left[\frac{\sum_{k=1}^{l}\omega\mathrm{k}}{\omega i}\mathrm{Wi}[ts,te]\right] + \left[\frac{\sum_{k=1}^{l}\omega\mathrm{k}}{\omega min} + \frac{\sum_{k=1}^{l}\omega\mathrm{k}}{\omega i} + (l-1)\right]\mathrm{M} \tag{10}$$

In equation 10, the section $\frac{\sum_{k=1}^{l}\omega\mathrm{k}}{\omega i}\,\mathrm{W}i[ts,te]$ represents total sending bits from a perfect, fair and ideal system with W*i[ts,te]* bits from i^{th} flow in worst and smooth condition.

4.4 The complexity of CBLLFQ scheme

The complexity factor of CBLLFQ scheme is measured by packet enqueuing and dequeuing time. The enqueuing process involves queue size calculation and packet enqueuing or dropping. According to PQ and DWRR the packet dequeuing process from the four queues consists of token preemption and actual packet dequeuing. In CBLLFQ, the real-time voice and video traffic have more priority than other traffic types as discussed previously. So, the real-time traffic is not affected in heavy congestion; whereas the WFQ doesn’t involve such type of customization, therefore the complexity of WFQ is very high O(N) [28]. The proposed scheme executes all dequeuing steps in a constant time, so the time complexity of CBLLFQ scheme is O(1). To compute the complexity, the algorithm should keep track the quantum size Q(t) and queue size calculation time V(t). The Q(t) and V(t) increase linearly by simply increment of one per unit time. The time is updated after each “K” events of arrival and departure of packet “P”. The $Q(t_{k-1})$ and $V(t_{k-1})$ is the time at the event (k-1). Then according to equation 11, the Q(tk) and V(tk) will be:

$$Q(t_k) = \mathrm{Max}(\,Kp,\ Q(t_k))$$
$$V(t_k) = \mathrm{Max}(\,Kp,\ V(t_k)) \tag{11}$$

And the Q(t) and V(t) at any time “t” will be:

$$Q(t) = (t - t_{k-1}) + Q(t_{k-1})$$
$$V(t) = (t - t_{k-1}) + V(t_{k-1}) \tag{12}$$

Now according to equation 12, the Q(t) and V(t) is the sum of all backlogged “Q” and “V” respectivily at event “K” of packet “P”.

5. Simulation Setup and Performance Evaluation

In this section, the simulation setup and QoS parameters like throughput, delay, packet loss, and fairness of CBLLFQ algorithm are analyzed. The model is simulated and evaluated using Network Simulator NS-2. The concentration is on how the real-time traffic is affected by non-real time traffic in different scenarios. Voice and video packets are mapped to UMTS classes in edge routers. The queuing, scheduling and policing mechanisms are implemented in the core routers of the UMTS network. The three basic components of the simulation, as shown in **Fig. 3**, are Terminal Equipment (TE) or User Equipment (UE), UTRAN and UMTS core network. The functionality of the core network depends on different routers.

In the simulation setup, there are core routers, edge routers, and four application servers. The application servers provide different services according to the user'sservice requirements like, video, voice, FTP and web. Application servers send data packets to the edge routers like, Egress and Ingress routers. Through these routers, a defined DiffServ Service Code point (DSCP) is assigned to the data packets based on the type of application and then sent to GGSN (core router). The GGSN router differentiates each flow of IP data packet according to its DSCP value, which remains same as assigned from the external network.

After this, the IP data packets are transmitted to the SGSN router with specific scheduling and queuing priorities. After receiving packets, the SGSN router transmits these packets to the RNC, where the packets of IP data are transformed into RLC SDU. Next, the DCH is used with acknowledge mode. The maximum transmission time for the RLC layers is unlimited. These all configurations are given in EURANE [23] standard. The CBLLFQ scheduling algorithm is very flexible as it is based on different types of traffic and can modify the priority of each class. The core parameters for calculating the performance are IP data packet loss, delay, jitter and bandwidth of the link. The link between each functional unit uses 1 Mbps defined bandwidth, whereas different types of protocols and traffic characteristics are selected based on the traffic classes as mentioned in **Table 1**.

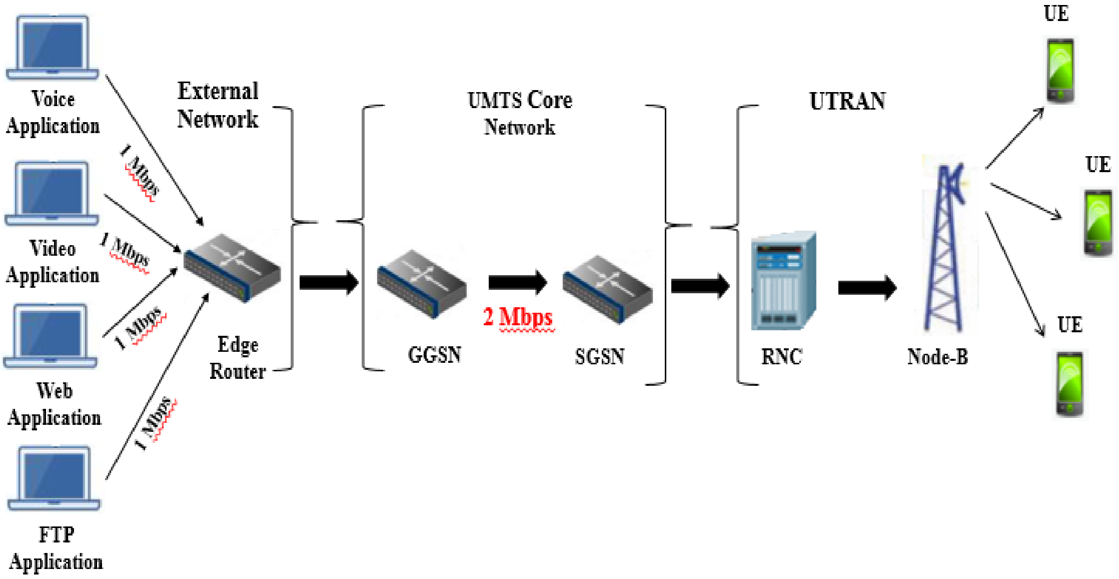

Fig. 3. Experimental Setup for End to End QoS Provisioning

Minimum and maximum threshold values for green, yellow and red traffic for RED queuing mechanism are as: (0 , 50), (50 , 100) and (100 , 150). For the both smooth and congestion traffic scenario the QoS investigation is performed against the existing scheduling algorithm like WRR and LLQ and the proposed one i.e. CBLLFQ. The traffic load between the links is tested with 2 to 25 mobile users' nodes. Among the whole traffic the conversational class packets are allowed to pass; which are followed by the traffic of video streaming, interactive and background classes based on weights. The voice packets in PQ is allocated with the highest level of priority, while least level of priority is assigned to the background class. The weights of different classes are discussed in the proposed algorithm section. The communication cost is reduced w.r.t delay, throughput and packet loss; because a 2 Mbps core section's link bandwidth is efficiently managed by the CBLLFQ for both real-time and non-real time traffics in normal and heavy traffic cases. This helps to improve QoS in the end to end communication. To obtain scalability, the following cases are observed:

Case1: If the concern is to send more real-time traffic then the size of the queue is to increase; which is controlled by the PQ scheduler. For this, one need to increase link bandwidth of the end to end network. The queue size will be increased by installing a physical queue and configuring it.

Case2: If the concern is to handle more non-real time traffic then processes in case 1 is repeated for the queues, but is handled by DWRR scheduler.

Case 3: If we want to send more real time and non-real time traffic then processes in both case 1 and 2 are used, but an improved process capability of the routers is needed too.The simulation scenario is shown in **Fig. 3**, with the investigated parameters, which are shown in **Table 2**.

Table 2. Simulation Parameters

Parameters	Values
Link bandwidth	1 Mbps
DCH bandwidth	384Kbps
GGSN-SGSN bandwidth	2 Mbps
Radio link RLC PDU error	Uniform mean=0.01
Frequency	10 MHz
Access Time	1 ms
Active UE number	2 - 25
No of Simulation	20
Fast Power Control	Ideal
CTR	1
PTR	10
Quantum size	1000 Bytes

6. Results

This simmultion based research evaluates delay, throughput, packet loss and fairness for VoIP and video traffics. A different scheduling algorithm like Weighted Round Robin (WRR), Low latency Queuing (LLQ), and (CBLLFQ) are implemented; to handle VoIP and video traffics. We observed that the throughput of the proposed CBLLFQ algorithm is better than other existing scheduling algorithms. In the case of video traffic, during less load , the throughput of CBLLFQ is equal to LLQ, but remains better against the heavy load. Due to the variable size of packets, the WRR shows worst performance against video traffic.

The main characteristic of the scheduleris to provide a bounded delay to the traffic. The delay of the proposed scheduler is better than others against normal and heavy traffic. Following **Fig. 4** to **10**. show a comparison betwee existing WRR and LLQ algorithms, and our proposed one i.e. CBLLFQ. The **Fig. 4** and **5** compare them based upon the delay factor against VoIP and video traffics. We can see that the CBLLFQ is reducing its delay using class-based queuing management because a PQ is used for the voice traffic and less complex DWRR is used for other traffics. In both types of traffics due to the unfairness factor, the WRR does not provide any upper bound guarantee for the delay and has worst performance than others.

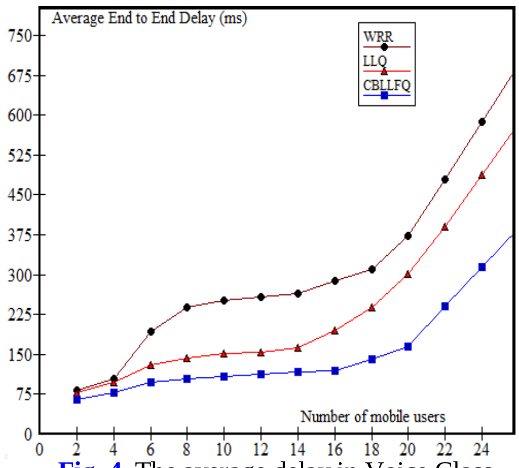

Fig. 4. The average delay in Voice Class

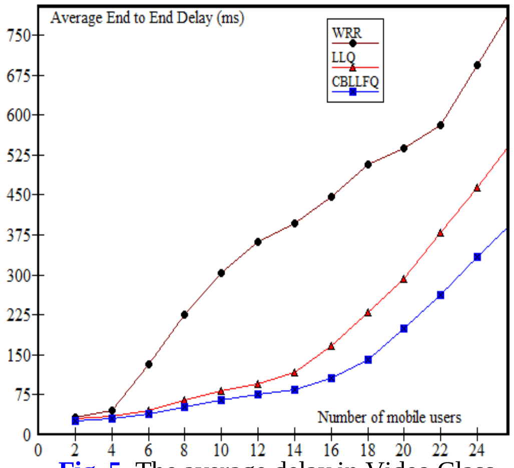

Fig. 5. The average delay in Video Class

The delay in VoIP traffic increases rapidly because there is no priority for voice traffic. As presented in **Fig. 6** and **7**, the throughput is increased because high priority is the cause of transferring a maximum number of voice packets between GGSN to SGSN bottleneck. The proposed algorithm increases the fairness along side the throughput of video traffic is also increased for both normal and worst traffic cases. Incase of heavy traffic, the throughput of WRR and LLQ is decreasing rapidly.

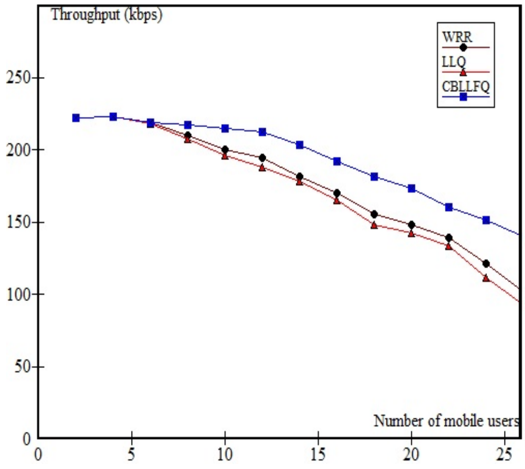

Fig. 6. Throughput in Voice Class

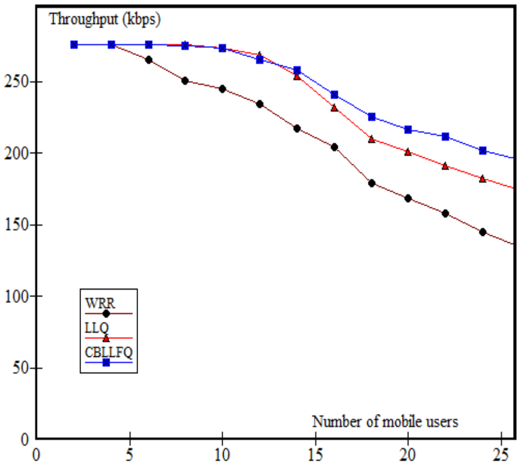

Fig. 7. Throughput in Video Class

The combination of the weighted distribution of network bandwidth in DWRR and PQ played an important role to provide fairness property in different traffic classes. The Fig. 8 shows that the obtained fairness characteristics of CBLLFQ are better than LLQ. The packet loss rate, of the proposed algorithm, as depicted in Fig. 9 and 10, is very low than other algorithms for normal and high traffic. The separate queues for voice and video traffics with PQ and DWRR mechanism make this possible for the CBLLFQ algorithm to keep packet loss rate within the desired limit of a selected QoS's standard. The CBLLFQ is controlling the packet loss rate in case of low, high and very high traffic. Due to the selection of best quantum and weight values, the average packet loss rate is reduced, whereas the packet loss rate of WRR is very high for both type of traffics. Hence our customized queuing mechanism is better which can handle the packet loss issue efficiently.

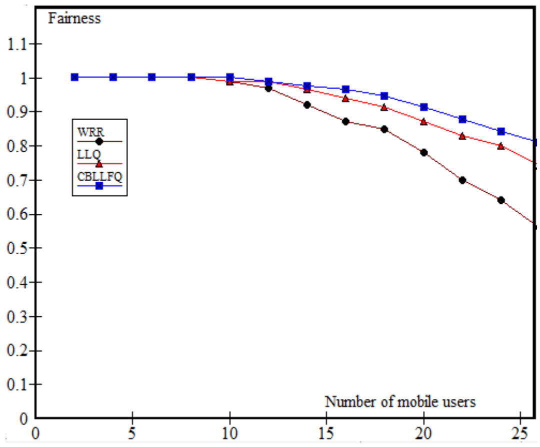

Fig. 8. Fairness Levels in Schedulers

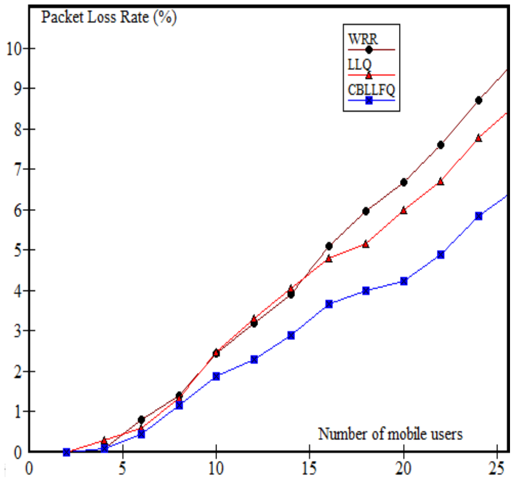

Fig. 9. Average Packet Loss Rate in Voice Class

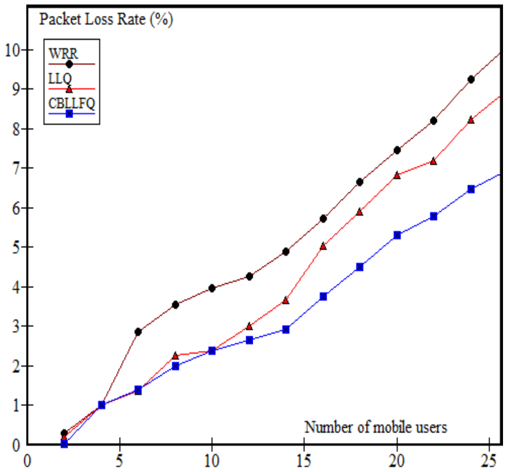

Fig. 10. Average Packet Loss Rate in Video Class

7. Application Perspective

HSDPA allows networks based on UMTS to have higher data speeds and capacity to maximize system performance by using various adaptive techniques, while satisfying QoS constaints. In the future a data rate of upto 10-20 Mbps will be required for several applications such as, real-time streaming voice or video. HSDPA promises a data rate of up to 10Mbps, five times greater than that of 3G/UMTS [30]; in latest Ubiquitous technologies, such as mobile phones, radio-frequency identification tags, GPS, and interactive white boards.With respect to the economic growth in both mobile and Internet industry, as well as in the high technology markets in general, have triggered new research challenges to address the commercial provisioning of mobile data services, including 3G, 3.5G etc. There are around 2.8 billion global WCDMA (including WCDMA + HSPA) Subscribers worldwide, where as there are 2.0 billion HSPA subscribers [31]. UMTS already reported during 2015 that 3G/4G subscriptions has already passed 3 billion milestone [32]. In 2015, Ericson's mobility reports predicts that the mobile data traffic over the 6-year period from 2015-2020 will equal 860 extrabytes - comparied with just 70 extrabytes over the previous 6 years from 2009-2014 [32]. The percentage split of WCDMA (Family) / HSPA connections by region are depicted in the following **Table 3** [31]. Africa and Asia Pasific has increased its market growth rate with 7% and 6% repectively [31].

Table 3. Worldwide Connection Distribution of HSPA

Region	Percentage WCDMA	Percentage HSPA
Africa	64%	14%
Asia Pacific	34%	46%
Latin America	2%	20%
Europe: Western	0%	15%
North America	0%	5%

Networked multimedia applications are also big consumers of the broadband services of HSPA/HSDPA with either real-time (e.g. voice, video streaming communication) and/or non real-time (e.g. FTP, Web) requirements [31, 33]. Third generation systems demands good bandwidth as a basic need in combination with global mobility and interactivity; hence enables new applications and opens new business areas. Among these video telephony and video conferencing are most aspiring telecommunication services. The availability of high bandwidth is now emerging in both fixed and mobile telecommunications networks, and our proposed algorithm supports such networks in classification, queueing and schdeuling the network traffic in a more robust manner.

According to the cummulative facts of Statista [34], about the users worldwide from 2005 to 2018, that the number of internet users are 3.9 billion, up from 3.65 billion in the previous year.The economic growth has excelled with billions of dollors and have enabled the growth of thousands of telecommunication companies; moreover the businesses, government and several other organizations have also moved towards establishing e-business platforms. The ubiquitous characteristic of mobile telephony have the potential for accommodating device roaming and mobility. IP has become an end-to-end protocol of the delivery of most healthcare services, monitoring services and entertainment services worldwide; because IP exists both in the wireline as well as in the wireless world, in offices environments and smart home networks [33]. Internet essentially provides a mechanism for re-routing packets from one part of the IP network to another, along side mobile IPv6 is well suited for cellular networks for a number of reasons, such as availability of enough addressing space etc. Hence resulted into vast usage of cellular networks which provide 3G/4G services.

8. Conclusion

The main contribution of this research is to introduce an optimized scheduling algorithm which has several goals such as, to provide less delay, less packet loss, less complexity and high throughput with fairness properties. The proposed CBLLFQ algorithm mapps the DiffServ classes with UMTS's QoS parameters and assigns different queuing mechanism according to the marked DSCP codes.The scheduling is performed using a combination of PQ and DWRR algorithms along with our proposed CBLLFQ algorithm. We simulated the network to evaluate and compare the performance of CBLLFQ scheduling algorithm for different traffic scenarios and different algorithms like WRR and WFQ. The mathematical representation of an important system's factors is identified, formulated and calculated. The simulation results prove that the scheduling algorithm reduces the ratio of delay, packet loss to provide better utilization of network link; and keeps all these quality factors within the standard limitation of a choosen QoS standard. The algorithm results in fair sharing of the resource between different types of traffic such as voice, video, FTP and web. The algorithm is fair even during traffic congestion in links and increases the capacity of the overall system. Using DWRR, the complexity level is reduced in our proposed algorithm.

References

[1] A. M. Alkharasani, M. Othman, A. Abdullah, and K. Y. Lun, "An optimized aggregate marker algorithm for bandwidth fairness improvement in classifying traffic networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, pp. 1-14, Mar. 2016. Article (CrossRef Link).

[2] J. Jin, M. Palaniswami, D. Yuan, Y. N. Dong and K. Moessner, "Priority Service Provisioning and Max-Min Fairness: A Utility-Based Flow Control Approach," *Journal of Network and Systems Management*, vol. 25, no. 2, pp. 397-415, Apr. 2017. Article (CrossRef Link).

[3] J. Khamse-Ashari, G. Kesidis, I. Lambadaris, B. Urgaonkar, and Y. Zhao, "Max-min Fair scheduling of variable-length packet-flows to multiple servers by deficit round-robin,"in *Proc. of Annual Conference on Information Science and Systems (CISS), Princeton, NJ*, pp. 390-395, 2016. Article (CrossRef Link).

[4] N. M. Madi, Z. M. Hanapi, M. Othman and S. Subramaniam, "Two-level QoS-aware frame-based downlink resources allocation for RT/NRT services fairness in LTE networks," *Telecommunication Systems, Springer*, vol. 66, no. 3, pp. 357-375, Feb. 2017. Article (CrossRef Link).

[5] T. H. Szymanski, "An Ultra-Low-Latency Guaranteed-Rate Internet for Cloud Services," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 123-136, Feb. 2016. Article (CrossRef Link).

[6] C. H. Fu, Y. Y. Yen, "A Study on a Differentiated Service Queuing Scheme Based on Transmitted Bytes Within a UMTS Core Network Gateway," *Wireless Personal Communications*, Springer, vol. 68, no. 4, pp. 1539-1563, Feb. 2013. Article (CrossRef Link).

[7] S. Przylucki and D. Czerwinski, "Priority-Aware Packet Pre-marking for DiffServ Architecture Based on H.264/SVC Video Stream Structure," *Wireless Personal Communications, Springer*, vol. 96, no. 4, pp. 5391-5408, Oct. 2017. Article (CrossRef Link).

[8] G. Tychogiorgos and K. Leung, "Optimization-based resource allocation in communication networks," *Computer Networks*, vol. 66, pp. 32-45, Jun. 2014. Article (CrossRef Link).

[9] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round robin," *IEEE/ACM Trans. Networking*, 4 (3), 375-385, 1996. Article (CrossRef Link).

[10] W. Abbas, N. Abbas, and U. Majeed, "Performance Enhancement of End-to-end Quality of Service in WCDMA Wireless Networks," *Science International (Lahore)*, vol. 26, no. 2, pp. 613-620, Feb. 2014. Article (CrossRef Link).

[11] F. Checconi, L. Rizzo and P. Valente, "QFQ: Efficient packet scheduling with tight guarantees," *IEEE/ACM Trans. Networking*, 21(3), 802-816, 2013. Article (CrossRef Link).

[12] S. Karamchati, S. Rawat and V. Varma, "A novel architecture to enhance Quality of Service in IP networks," in *Proc. of International Conference on Information Networking (ICOIN), Da Nang*, pp. 616-621, 2017.

[13] R. Sharma, N. Kumar, and S. Talabattula, "Performance of new dynamic benefit-weighted scheduling scheme in DiffServ networks," in *Proc. of International Conference on Advances in Computing, Communications, and Informatics (ICACCI), New Delhi*, pp. 2578-2583, 2014. Article (CrossRef Link).

[14] M. Alreshoodi, J. Woods and I. K. Musa, "QoE-enabled transport optimization scheme for real-time SVC video delivery," in *Proc. of 9th International Symposium on Communication Systems, Networks & Digital Sign (CSNDSP), Manchester*, pp. 865-868, 2014. Article (CrossRef Link).

[15] R. Deng, J. Yang, C. Liu, and G. Liu, "A delay prediction-based content and channel aware packet scheduler for real-time video over HSDPA," in *Proc. of IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks," (WoWMoM), Madrid*, pp. 1-6, 2013. Article (CrossRef Link).

[16] Simulation of Enhanced UMTS Access and Core Networks (SEACORN) Project, online: https://www.it.pt/Projects/Index/239

[17] "Quality of Service (QoS) Concept and Architecture," TS 23.107 version 10.2.0 Release 10, 2011. Online:https://www.etsi.org/deliver/etsi_ts/123200_123299/123207/10.00.00_60/ts_123207v100000p.pdf

[18] N. Shetty, G. Schwartz,and J. Walrand, "Internet QoS and Regulations," *IEEE/ACM Transactions on Networking*, vol. 18, no. 6, pp. 1725-1737, Dec. 2010. Article (CrossRef Link).

[19] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group," *IETF RFC 2597*, June 1999. Article (CrossRef Link).

[20] K. Nichols, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," *IETF RFC 2474*, Dec. 1998. Article (CrossRef Link).

[21] H. Su, M. Atiquzzaman, "ItswTCM: a new aggregate marker to improve fairness in DiffServ," in *Proc. of IEEE GLOBECOM*, pp. 1841-1846, 2001. Article (CrossRef Link).

[22] R. Braden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture: An overview," *IETF RFC 1633*, Jun. 1994. Article (CrossRef Link).

[23] Enhanced UMTS Radio Access Network Extensions (EURANE) for NS2, online: https://www.scribd.com/document/91491046/Enhanced-UMTS-Radio-Access-Network-Extensions-for-Ns-2

[24] L. Xiuqin, Y. Luoheng and L. Julong, "A DiffServ Supporting and Distributed Dynamic Dual Round Robin Scheduling Algorithm on DS-CICQ," *Physics Procedia*, vol. 33, pp. 804-810, 2012. Article (CrossRef Link).

[25] T. Demoor, J. Walraevens, D. Fiems, and H. Bruneel, "Performance analysis of a priority queue: expedited forwarding PHB in DiffServ," *AEU-International Journal of Electronics and Communications*, vol. 65, no. 3, pp. 190-197, 2011. Article (CrossRef Link).

[26] K. Nichols, V. Jacobson, and K. Poduri, "An Expedited Forwarding PHB Group," *RFC2598 IETF Diffserv Working Group*, June 1999. Article (CrossRef Link).

[27] "High-Speed Downlink Packet Access (HSDPA): Overall Description," TS 25.308 version 13.2.0 Release 13, 2016. Online https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1171

[28] D. S. Lee, C. M. Chen and C. Y. Tang, "Weighted Fair Queueing and Compensation Techniques for Wireless Packet Switched Networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 1, pp. 297-311, Jan. 2007. Article (CrossRef Link).

[29] H. Alnuweiri and H. Tayyar, "Analysis of virtual-time complexity in weighted fair queuing," *Elsevier Computer Communications*, vol. 28, no. 7, pp. 802-810, 2005. Article (CrossRef Link).

[30] D. Singhal and N.Jotwani, "A fair packet scheduler to support QoS in HSDPA," in *Proc. of First International Communication Systems and Networks and Workshops, IEEE*, pp. 1-7, 2009. Article (CrossRef Link).

[31] Fast Facts, 2015 UMTS Forum, online: https://www.umts-forum.orgcontent/view/2001/281/

[32] C. Solbé, "3G/4G subscriptions pass 3 billion milestone", online: https://www.umts-forum.org/content/view/4655/303, retrieved on = August 2019.

[33] R.C. Qiu, W. Zhu and Y.Q. Zhang, "Third-generation and beyond (3.5 G) wireless networks and its applications," in *Proc. of IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No. 02CH37353)*, vol. 1, pp. I-I, 2002. Article (CrossRef Link).

[34] J. Clement, "Number of internet users worldwide from 2005 to 2018 (in millions)," online: https://www.statista.com/statistics/273018/number-of-internet-users-worldwide, retrieved on = October 2019.

Sohail Ahmed received the B.S. degree in computer engineering from COMSATS University, Islamabad, Pakistan in 2010 and M.S. degree in computer engineering from University of Engineering & Technology, Taxila, Pakistan in 2015. Since 2016 he has been a lecturer with the Department of Computer Science & Information Technology, University of Lahore. Before this, he has served PTCL, a cellular company in Pakistan, from 2005 to 2015. His research interest includes Internet services, Optimization, algorithm and design, Internet of Things, digital design and Embedded System.

Malik Muhammad Asim received B.S. and M.S. from University of Engineering and Technology (UET), Taxila, Pakistan. He is doing Ph.D. in computer engineering from UET, Taxila. Since 2011 he has been an Assistant Professor with Department of Computer Engineering, University of Engineering & Technology. Before this, he was a lecturer in the Computer Engineering Department from 2005 to 2011. He has more than ten years of research and development experience. He is also a postgraduate student advisor. His research interest includes computer networks, distributed system, Multimedia Systems, image processing.

Nadeem Qaisar Mehmood received his Ph.D. from University of Camerino, Italy in 2016. He is currently an Assistant Professor at CS and IT department, of University of Lahore, Gujrat Campus, Pakistan. His researches in the domains of smart environments, artificial intelligence, emerging technologies, real-time web, big data and Internet of Things(IoT). He has a M.Sc. in Computer Science from PUCIT, Pakistan and a M.S. degree in Data and Knowledge Engineering from Otto-von-Guericke University, Germany in 2004 and 2008 respectively.

Mubashir Ali received the M.S. degree from Bahria University, Islamabad, Pakistan, in 2014. He is currently an Assistant Professor with the department of Computer Science and Information Technology, University of Lahore, Pakistan. He has seven years of software development experiencein research and development-based publicsector organizations in Pakistan. He has authored over ten publications in international journals andconferences. His active research areas are NLP, Data mining, machine learning, software repository mining, and business intelligence.

Babar Shahzaad received B.S. and M.S. from University of Engineering and Technology, Taxila, Pakistan. Since 2016 he has been a lecturer with the department of Computer Science, University of Management &Technology. His research interest includes Android Development, Internet of Things (IoT), Information-Centric Networking (ICN), Embedded Systems.