An optimal bandwidth allocation algorithm for improving QoS in WiMAX

Zeeshan Ahmed, Salima Hamma, Zafar Nasir

Noname manuscript No.
(will be inserted by the editor)

An Optimal Bandwidth Allocation Algorithm for Improving QoS in WiMAX

Zeeshan Ahmed · Salima Hamma · Zafar Nasir

the date of receipt and acceptance should be inserted later

Abstract In the last few years, the term "quality of service" has become increasingly synonymous with digital cellular networks and has greatly influenced the way we manage network traffic. The IEEE 802.16 standard is a broadband wireless access system that enables high speed data transfer over large distances. It is one of the standards that meet the IMT-Advanced specifications. It also incorporates a quality of service framework to provide quality of service to both realtime and non-realtime multimedia applications. One of the critical contributions of a QoS framework is efficient scheduling of network traffic. This paper dilates on a two-level scheduling algorithm for base station uplink scheduler to provide quality of service to various classes of traffic. The proposed algorithm ensures efficient and fair multimedia transmission. We also deliberated on a video transmission framework based on the proposed algorithm. The performance of two-level scheduling algorithm has been extensively analyzed through simulations and the results have effectively established the efficacy of the proposed algorithm. The results reveal that the the proposed algorithm is able to fairly and efficiently schedule network traffic while ensuring quality of service for all classes of traffic.

Keywords IEEE 802.16 · WiMAX · QoS · Packet scheduling · Bandwidth Allocation

Zeeshan Ahmed
FAST-NUCES University, Shah Latif Town, Karachi, Pakistan.
Tel: +92-345-3013846
E-mail: zeeshanahmed@nu.edu.pk

Salima Hamma
Laboratoire des Sciences du Numerique de Nantes (LS2N), Universite de Nantes, 2 Chemin de la Houssiniere, 44322 Nantes, France.
E-mail: salima.hamma@univ-nantes.fr

Zafar Nasir
Indus University, Gulshan Iqbal, Karachi, Pakistan.
E-mail: zafarnasir@indus.edu.pk

1 Introduction

In the last two decades, the number of computer and mobile phone users have increased manifold. Moreover, we have also witnessed multifold increase in the usage of multimedia services, such as VoIP, IPTv, and video conferencing. These services require much more network resources as compared to simple data services. Furthermore these services have more stringent quality of service (QoS) requirements. Therefore, there is a need of efficient and more capable networks to support these and future services. In this regard IEEE 802.16 broadband wireless access (BWA) standard [1] is an excellent choice. The standard is commercially known as WiMAX, which stands for Worldwide Interoperability for Microwave Access.

WiMAX Forum [2] describes WiMAX as "a standards-based technology enabling the delivery of last mile wireless broadband access as an alternative to cable and digital subscriber line (DSL)". WiMAX offers high speed data transfer over long distances for both stationary and mobile users. Furthermore it incorporates an extensive QoS framework to support different classes of traffic. A WiMAX point-to-multipoint (PMP) network is a digital cellular network in which a base station (BS) manages and furnish services to multiple subscriber stations (SS). An SS is an equipment that allows end-user to communicate with a BS. The BS then provides connectivity to core network, as shown in Fig. 1.

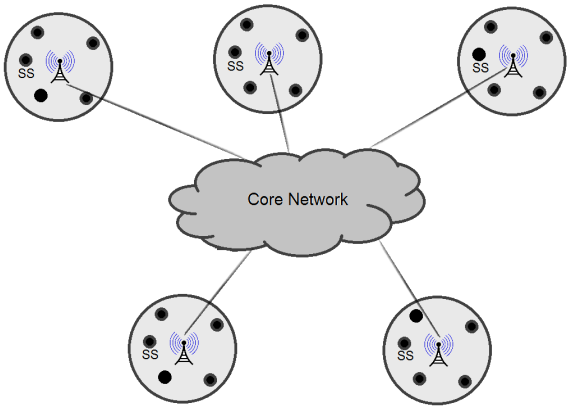


Fig. 1 A PMP WiMAX Network

To ensure satisfactory transmission of both multimedia and data traffic, a network must provide sufficient level of QoS to all types of traffic. However providing QoS in wireless networks, while ensuring privacy and security, is much more challenging as compared to wired networks. This is due to the unpredictable nature of wireless medium and mobility of SS. During propagation through the wireless medium radio waves encounter multiple impairments such as multi-path propagation, interference, and attenuation. Both QoS and network security are active areas of research [3–7], however in this paper our focus will remain on QoS alone.

QoS controls manages network's data transmission by priotitizing time-sensitive and mission-critical services. Therefore data transmission management mechanisms, such as call admission control and packet scheduling, lie at the heart of a QoS framework. Network traffic scheduling based on differentiation of traffic classes is one of the most important and essential functionalities of a QoS architecture . In this context, a scheduler decides the timing and order of transmission of data packets so as to ensure QoS for all service flows. The complex task of scheduling is performed by three distinct schedulers in WiMAX: BS uplink scheduler, BS downlink scheduler, and SS scheduler. Downlink scheduling is relatively simple as the BS is the only transmitter in the downlink direction. While the uplink scheduling is much more challenging as the BS uplink scheduler must synchronize its decision with all SSs.

BS uplink scheduler is responsible for scheduling packets from SS to BS. These data packets are stored in queues that are maintained at SS. These queues are not directly accessible by BS uplink scheduler and so the scheduler cannot determine the exact sizes and deadlines of the stored packets. Therefore the uplink scheduler has to make decisions according to estimates. The functions of each scheduler are well-defined by the standard. However, the mechanisms to achieve this functionality have not been defined by the standard. Therefore vendors and service providers can choose the scheduling schemes that best suit their needs.

The standard provides support for both realtime and non-realtime traffic. Realtime traffic is divided into constant bit-rate (CBR) traffic and variable bit-rate (VBR) traffic. The scheduling of CBR realtime traffic is straightforward and well-defined by the standard. However, the scheduling algortihms for VBR realtime and non-realtime traffic are not defined in the standard. Scheduling VBR realtime traffic is the most challenging among all classes of traffic due to its bursty nature and tight delay constraints [8]. Therefore the scheduler must make sure that the packets are delivered before the deadlines are expired, otherwise they may be of no value to the receiver. Usually, applications such as video conferencing, and IPTv etc. use VBR realtime services. These applications can tolerate some degree of lost packets. However, if many packets miss their deadline and loss become significant, then it can seriously degrade the level of service as perceived by the end-user. Therefore, this service type is given priority over non-realtime traffic. The scheduler must also make sure that lower priority classes also get acceptable level of service and no connection starve even under high load.

In this paper we have proposed a two-level QoS-aware packet scheduling algorithm (TLSA) for BS uplink scheduler and a video transmission framework by extending our work on intra-class scheduling algorithm for VBR realtime class [9] and intra-class scheduling algorithms for non-realtime VBR and best effort classes [10]. At the first level uplink bandwidth is distributed among different service classes, and then at the second level intra-class bandwidth distribution is done. The objectives of TLSA are as follows: (i) To provide QoS to all classes of traffic (ii) To fairly allocate resources among connections within each service class (iii)To ensure that lower priority flows would not affect higher priority flows (iv) To prevent starvation of lower priority flows (v) To ensure high resource utilization.

The remainder of the paper is organized as follows. Section 2 provides details of QoS architecture provided by the standard. Then, Section 3 gives an overview of the related work. In section 4 we provide the details of TLSA. In the next section, we present our video transmission mechanism. In section 6 simulation results are provided to show the performance of the proposed solution. Section 7 concludes the paper.

2 QoS Architecture Provided by WiMAX for Point-to-Multipoint Networks

The WiMAX system has four key layers: Media Access (MAC) convergence, MAC sublayer, MAC privacy, and the physical layer. The MAC sublayer is responsible for providing QoS in WiMAX. WiMAX MAC is connection oriented, i.e. a connection must be established between an SS and a BS before any transmission could occur. A connection can be initiated either by an SS or by a BS. Each connection is identified by a unique 16-bit connection identifier (CID). A connection could be used to manage multiple service flows. A service flow can be defined as a sequence of packets in one-way direction, which are characterized by same QoS parameters, i.e. it is a unidirectional flow of packets that is provided a particular QoS. Each service flow is identified by a unique 32-bit identifier, called service flow identifier (SFID).

A service flow can be in one of the following states: provisioned, admitted, or active. Each state has an associated set of QoS parameters. These parameters are set of quantitative service measurements such as minimum bandwidth, maximum delay, jitter, and maximum packet loss rate. An incoming service flow enters the provisioned state. However, no data transfer could occur until it is switched to active state. Once the QoS parameter set for admitted state (AdmittedQoSParamSet) or active state (ActiveQoSParamSet) become known, the service flow could be switched to admitted or active state. *AdmittedQoSParamSet* defines QoS parameters for which the system is reserving resources. The main resource to be reserved is bandwidth. *ActiveQoSParamSet* is the set of QoS parameters actually being provided to the service flow.

For a new service flow, the call admission control (CAC) module in the BS analyzes the requested QoS parameters and determine whether the request could be fulfilled or not. If the available resources are sufficient to fulfill the requested QoS, then the BS assigns a unique SFID to the service flow.

WiMAX supports both frequency division duplex (FDD) and time division duplex (TDD). Furthermore, the standard uses orthogonal frequency division duplex (OFDM) to efficiently share the medium among SSs. Thus, it can operate as either FDD/OFDM or TDD/OFDM. Since, majority of applications make asymmetric use of bandwidth, therefore TDD is the preferred duplexing mode. In TDD, a MAC frame is divided into uplink and downlink subframes. The duration of these frames can be dynamically adjusted by the BS according to traffic conditions. A TDD frame is shown in Fig. 2.

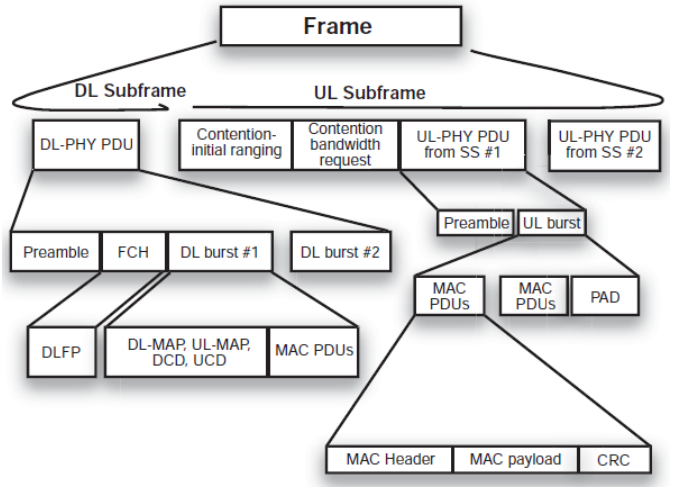


Fig. 2 Structure of a TDD MAC Frame [11]

In the downlink, BS is the only transmitter. Within a given frequency band all SSs receive same transmission. A downlink subframe contains one or more bursts for individual SSs. The DL-MAP field in downlink subframe defines which burst is designated for which SS. Similary, UL-MAP message specifies which SSs can transmit to the BS in each burst.

There are three schedulers incorporated in the standard: BS uplink scheduler, BS downlink scheduler, and SS scheduler. The BS uplink scheduler decides which SS can transmit data to BS at a particular time. BS downlink scheduler controls the transmission from BS to SSs i.e. in the downlink direction. SS scheduler is responsible for distributing bandwidth, which is allocated to the SS by the uplink scheduler, among its active connections. The functions of these schedulers are defined, however their working is not defined by the standard. Therefore vendors and service providers are free to choose any scheduling scheme that fulfill their requirements. The QoS architecture provided by the standard is shown in Fig. 3.

To support different types of applications, five scheduling service classes are provided by the standard: (i) *Unsolicited Grant Service (UGS):* UGS is designed to support realtime CBR applications, which generate fixed size packets periodically. For example, T1/E1 emulation and VoIP without silence suppression. (ii) *Realtime Polling Service (rtPS):* rtPS is designed for VBR realtime applications. These applications generate variable size data packets on periodic basis, such as audio and video streaming. (iii) *Extended Realtime Polling Service (ertPS):* The service is built on the efficiency of both UGS and rtPS. It is designed to support UGS like service flows which can become inactive for an interval. For e.g. VoIP with silence suppression. (iv) *Non-Realtime*

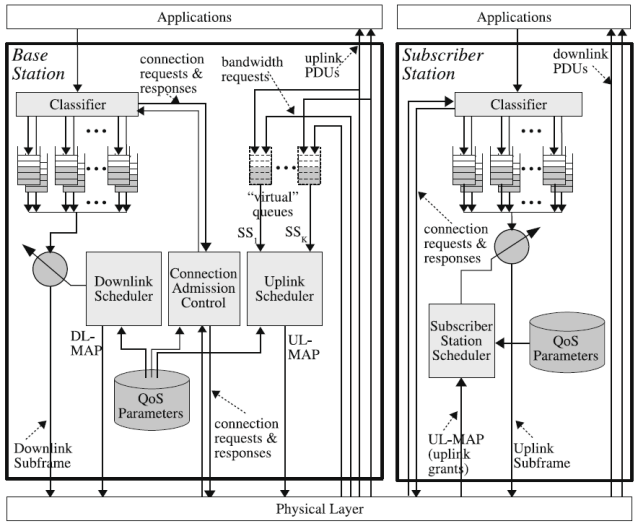


Fig. 3 QoS Architecture Provided by IEEE 802.16e [12]

Polling Service (nrtPS): The service is designed for delay tolerant services that generate variable size data packets on regular basis, such as file transfer protocol (FTP). (v) *Best-Effort (BE) Service:* BE is designed for applications that do not require any QoS, such as hyper text transfer protocol (HTTP). Table 1 specifies the QoS parameters associated with each class.

During initialization and network entry, the BS assigns up to three dedicated connections to an SS so as to provide the SS the ability to send and receive control messages. There are several ways an SS can then request bandwidth using the control connections, that includes request/grant mechanism, UGS allocation, unicast polling, multicast/broadcast polling, contention-based allocation, piggbacking. UGS flows get fixed amount of bandwidth periodically and therefore they do not need to explicitly request bandwidth. For an rtPS connection, the BS polls the SS to know its bandwidth requirements. rtPS flows cannot participate in contention process. While, nrtPS and BE connections can use contention-based mechanism during a contention period to request bandwidth. Furthermore, all traffic classes, except UGS, are allowed to make piggyback requests. The BS then allocate bandwidth, for all connections belonging to the SS, aggregated into a single grant. The SS scheduler is then responsible for distributing the grant among individual connections. The bandwidth request mechanisms available in 802.16 standard are summarized in Table 2.

A bandwidth request can be incremental or aggregate. In an incremental request the SS could ask for more bandwidth for a connection in an incremental fashion. While in an aggregate request the SS specifies the total bandwidth required for a connection. Most re-

Service Class	**QoS Parameters**	**Applications**
UGS	– maximum sustained traffic rate – minimum reserved traffic rate – delay tolerance – jitter tolerance	VoIP
ertPS	– maximum sustained traffic rate – minimum reserved traffic rate – delay tolerance – jitter tolerance	VoIP with silence detection
rtPS	– maximum sustained traffic rate – minimum reserved traffic rate – delay tolerance	Video Streaming
nrtPS	– maximum sustained traffic rate – minimum reserved traffic rate	FTP
BE	No QoS requirement	Web browsing

Table 1 QoS parameters associated with each service class

Type	**QoS Class**	**Mechanism**
Unsolicited Request	UGS & ertPS	Periodically allocates bandwidth at setup stage:
Piggybacking	ertPS, rtPS, BE & nrtPS	Piggyback request over any other MAC packets being sent to the BS
Bandwidth Stealing	nrtPS & BE	Sends BWR instead of general MAC packet
Contention region	ertPS, nrtPS & BE	Subscriber stations use contention regions to send bandwidth request
CDMA code-based request	nrtPS & BE	SS chooses one of the CDMA request codes from those set aside for bandwidth requests.
Unicast Polling	ertPS, rtPS, nrtPS & BE	BS polls each SS individually and periodically
Multicast & Broadcast Polling	ertPS, nrtPS & BE	BS polls a multicast group of SSs

Table 2 Bandwidth Request Mechanisms in IEEE 802.16 [13]

quests are incremental, however aggregate requests are periodically used so that the BS can update its perception of SSs bandwidth needs.

3 Related Work

Many researches have been done with the aim of proposing efficient scheduling schemes for WiMAX. Early researches proposed to use well-known algorithms such as Round Robin (RR) [14], Weighted Round Robin (WRR) [15] and Weighted Fair Queuing (WFQ) [16] for this purpose. However, these algorithms are generic in nature and do not take into account the details specific to WiMAX networks. Although simple, they are not efficient and effective for multi-class high speed networks.

The schemes proposed for WiMAX can be classified as either channel-unaware or channel-aware. Channel-aware schedulers make decisions according to current channel state information. Usually, these schemes give priority to SSs with good channel conditions. This results in efficient utilization of radio resources and thus system efficiency is enhanced. The main problem of this scheme is the unfair nature of allocation. SSs with poor channel conditions may starve for substantial intervals, while SSs with good channel conditions are usually over-provisioned. On the other hand, channel-unaware schemes focus on MAC layer mechanisms and assume ideal channel condition. The aim of these schemes is to guarantee QoS parameters such as minimum traffic rate, maximum delay bound and fairness.

Due to classes with different QoS requirements, a scheduler must impose some sort of priority order. Therefore, most of the proposed solutions are hierarchical in nature. Generally, these schemes first used inter-class scheduling algorithms to distribute bandwidth among different service classes. Then an intra-class scheduling algorithm, which may be different for different service classes, is used to distribute the allocated bandwidth within the service flows of the same class.

3.1 Inter-Class Scheduling

Inter-class scheduling algorithms are designed to distribute bandwidth among service flows of different classes while ensuring QoS for all classes. Many inter-class scheduling schemes have been proposed. Some researchers [17], [18], [19], [20], [21] have proposed strict priority disciplines for inter-class bandwidth distribution. Normally, the priority is set according to service classes i.e. UGS > ertPS > rtPS > nrtPS > BE. However, priority can also be set according to some other criteria such as backlog size or packet deadline. Lower priority flows only get bandwidth if some bandwidth is not utilized by higher priority flows. These schemes do no guarantee fair allocation of bandwidth. Furthermore lower priority flows could starve for long durations due to strict allocation.

To avoid starvation of lower priority flows, Chen et al. [20] has proposed Deficit Fair Priority Queuing (DFPQ) [22] to be used for inter-class scheduling. DFPQ also consider service class priority during decision making, but each service class is allowed a fix amount of bandwidth in each round. Safa et al. [23] argue that under this scheme critical realtime packets might lose their deadlines, so they propose to use Preemptive Deficit Fair Priority Queueing (PDFPQ) instead of DFPQ. They propose to set nrtPS and BE queues as preemptive, while rtPS queues are non-preemptive. Each non-preemptive queue can use a fixed amount of additional bandwidth to schedule packets that may miss deadline otherwise. Compared to the results presented in [24], the scheme provides slight improvement both in delay and throughput. However, the simulation is done for only four frames. It would be more interesting to perform the simulation for more frames.

X. Zhang et al. [25] has proposed the use of WFQ algorithm, as it can efficiently distribute bandwidth among realtime flows, while indirectly guarantees the delay. However, the algorithm ensures QoS for realtime flows only and QoS for non-realtime flows is not considered. In another study [26], Y. Shang and S. Cheng provide a hierarchical scheduling model in which Worst-case Fair Weighted Fair Queuing (WF^2Q) [27] is used for inter-class allocation. The weight for each class is equal to the sum of the minimum data rate of all the connections in that class. With dynamic adjustment of weight , the scheme can guarantee the minimum data rate and worst-case fairness. However, the main disadvantage is O(N) complexity and therefore the scheme may not be suitable for very high speed data networks. Cicconetti et al. [28] argue that fair queuing schemes are too complex to be implemented in WiMAX. They argue that latency-rate control algorithms are particularly suited for scheduling in WiMAX. WRR is also proposed by A. Sayenko et al. [29] for intra-class scheduling. They also propose that the BS should specify the order of slots so as to minimize the maximum jitter and delay. The main advantage of these schemes is their simplicity and O(1) complexity. However, how the weights are chosen is not defined by the authors. Another problem with WRR is that it can be unfair when all packets are not of the same size, which is the case in WiMAX.

In [30], Chan et al. has proposed a two-tier scheduling scheme. First, all connections are classified into following categories:

1. Unsatisfied: a connection is unsatisfied if the bandwidth allocated to it is less than its minimum requirement.
2. Satisfied: a connection is satisfied if the allocated bandwidth is between its minimum requirement and maximum requirement.
3. Over-satisfied: a connection is over-satisfied if the allocated bandwidth is more than its maximum requirement.

Then the first tier distributes bandwidth according to connections category. Bandwidth is first allocated to unsatisfied connections, then to satisfied connections and then to over-satisfied connections. The results show that the scheme is more fair and it could provide the MRTR for each connection. However the scheme does not distinguish among different classes of realtime traffic and therefore realtime and non-realtime connections are treated equally.

In [31], the authors propose a channel aware algorithm for scheduling in BWA systems. They argue that Delay Threshold Prioirty Queuing (DTPQ) is a good choice when both realtime and non-realtime traffics are present. Rather than choosing a fix delay, they select an adaptive threshold-based priority queuing scheme which consider both deadlines and channel state conditions for realtime users.

A token-bucket based scheduling mechanism is presented in [32] by T.C. Tsai and C.Y. Wang. To avoid starvation of lower-priority classes, they set a maximum bandwidth limit for each service class. When a service class gets more bandwidth than its threshold, its priority is decreased. The study does not study the fairness of allocation.

R. Fei et al. has proposed a dynamic bandwidth allocation algorithm [33]. They provide a utility function that considers the QoS requirements of each service class. Each class is assigned a weight, which is then used by the utility function to determine the optimal scheduling. The algorithm is designed for relay mode operation and it may not be efficient for point-to-multipoint networks.

Sengupta et al. has presented a scheme [34] of dynamically modifying MAC PDUs based on the feedback obtained about channel state through Channel Quality Indicator (CQI). A feedback mechanism present at the receiver's MAC layer gives feedback to transmitter which in turn changes payload of upper layer by aggregation or fragmentation. The dynamic modification of PDU size results in reduction in dropped and corrupted packets. Thus the system achieves higher throughput and lower end-to-end delays. The scheme does not differentiate in different service classes and no QoS guarantees are provided for realtime traffic.

In [35], authors proposed a self-adaptive scheduling (SAS) algorithm for base transceiver stations. The aim is to improve energy efficiency, reduce carbon emission, and develop a self-sustainable green cellular network. The algorithm controls the operating states of a BTS thereby exploiting the traffic loads of the BTS and the single-hop neighbor BSs thereof. Each active BS in this scheme independently and dynamically decides its operation state, thus resulting in a fully distributed system. Simulation results revealed that the proposed SAS algorithm can significantly increase the energy savings compared with existing protocols. The focus of SAS is energy efficiency rather than furnishing QoS to various classes of traffic.

3.2 Intra-Class Scheduling

3.2.1 rtPS

In the study [36], the authors applied different algorithms on rtPS traffic and provided comparative results. They consider RR, WRR, maximum Signal-to-Interference ratio (mSIR), and temporary removal scheduler (TRS) [37] in their study. The simulations results show that RR and WRR transmit the minimum number of packets and are very inefficient under medium to high load conditions. mSIR and combination of $TRS + mSIR$ deliver the maximum number of packets. However, they require large delays to deliver packets, which make them unsuitable for realtime applications such as VoIP and IPTv. The authors then present a modified version of mSIR, called mmSIR. mmSIR was able to reduce the end-to-end delay, but still the average delay is unacceptable for most realtime applications.

Some researchers ([32],[19],[24],[38]) suggest the use of Earliest Deadline First (EDF) for rtPS traffic. In [20] EDF is proposed for both uplink and downlink direction. Downlink traffic is given priority over uplink traffic. In [19], they propose to use the concept of arrival-service curve [39] to predict the arrival pattern of incoming rtPS packet. We provide some comments on the use of arrival-service curve in section 4. In this scheme, if enough bandwidth is not available then the bandwidth is distributed among all rtPS connections according to their average data rates. However, this distribution can actually result in some unused portions of bandwidth as shown by simulations in section 6.

In [29], the authors have proposed a single scheduler for all classes of traffic. They argue that scheduling disciplines like Fair Queuing (FQ) and EDF complicates scheduling and therefore they are not suitable for high speed networks. They further argue that the difficulty of accurately determining the deadlines of individual

packets stored in SS buffers and potentially unfair behavior of EDF makes it unsuitable for WiMAX networks. However, their scheme is less efficient than EDF for rtPS flows.

3.2.2 nrtPS and BE

J. Chen, W. Jioa, and H. Wang [20] propose WFQ for nrtPS flows and RR for BE flows. A similar scheme is proposed by K. Wongthavarawat and A. Ganz [19]. However, they propose that the bandwidth available for BE flows should be distributed equally among the BE flows. V. Rangel, J. Ortiz and J. Gomez [40] and DN Lai, TC Huang and HY Chi [41] also propose similar schemes. However, they have proposed First Come First Serve (FCFS) for scheduling BE class. In these scheme co-scheduling is done according to strict priority. Lower priority flows can only get bandwidth if some bandwidth is not utilized by higher priority flows. Therefore, these schemes can result in starvation of lower priority classes. Furthermore, these schemes do no guarantee fair distribution of bandwidth among flows of same service class.

A. Sayenko, O. Alanen and J. Karhula have proposed a scheme [42] similar to WRR. The scheme treats each connection as a separate session. The QoS requirements are used to determine the required number of frame slots, which then become the weights for WRR. The scheduling scheme comprises three stages (i) Allocation of minimum number of slots (ii) Allocation of unused slots (iii) Ordering of slots to improve the provisioning of QoS. The first stage is mandatory, while the other two are optimization steps. The calculation of number of slots for nrtPS class is done in the same way as that for rtPS class. The algorithm does not take into account the deadlines of rtPS packets.

The two-tier scheme [30] proposed by L. Chan, H. Chao, and Z. Chou classifies all connections into three categories: unsatisfied, satisfied, and over-satisfied. The algorithm calculates weight for each connection based on its category and the QoS parameters. The bandwidth is first allocated to unsatisfied connections, then to satisfied connections and then to over-satisfied connections. No distinction is made on the service classes of the flows. Therefore, it may be not be possible for the algorithm to ensure QoS for realtime applications.

4 Two Level Scheduling Algorithm

4.1 Terminology

Firstly, we present the terminology that is important to understand the rest of the article.

- r_i^{min}: minimum reserved traffic rate (MRTR) for connection i
- r_i^{max}: maximum sustained traffic rate (MSTR) for connection i
- d_i: delay limit for connection i (in terms of number of uplink subframes)
- br_k^i: bandwidth requested by connection i in frame k
- ba_k^i: bandwidth allocated for connection i in frame k
- n: number of connections admitted
- d_{max}: max(d_i), where i= 1,2,...,n
- $bTbl$: an $n \times d_{max}$ table to store rtPS bandwidth allocations
- r_k: unused bandwidth in frame k
- f: current uplink subframe
- SR_i: service ratio for connection i
- SR: total service ratio
- r^a: current value of available uplink bandwidth
- F_o: set of active connections of service class o

4.2 Call Admission Control

Call admission control (CAC) is a set of actions and permissions that permits or denies a connection to network on the basis of network ability [6]. When an SS sends a new connection request to the BS with a certain QoS parameters, the CAC determines whether the request can be accepted or not depending upon the requested QoS parameters and current network state. After accepting a connection request from an SS, the network has to ensure that QoS requirements of the new connection are met throughout the duration of the flow. Therefore admissibility of a new connection must be carefully determined so that the service guarantees can be provided to all active connections.

In TLSA, An incoming connection is admitted by the BS, if and only if the available bandwidth is sufficient to guarantee the MRTR for the connection. Mathematically, a connection i is admitted if and only if $r_i^{min} \leq r^a$. After admitting i the value of r^a is updated, $r^a \leftarrow (r^a - r_i^{min})$. Thus, the CAC in TLSA is "without degradation", i.e. no degradation in QoS of existing connection is permitted to accomodate a new connection.

A BE connection has no MRTR and therefore it is always admitted. To ensure that connection i never surpasses its contract, it is assumed that a traffic limiting module is present at the SS that always keeps the bandwidth demands of i below r_i^{max}. Thus the traffic generated by connection i always remain between r_i^{min} and r_i^{max}.

4.3 First Level Scheduling

As the name suggests, the proposed scheduling scheme comprises two levels. The first level scheduling (FLS) algorithm distributes available uplink bandwidth among different service classes according to their bandwidth demands and QoS requirements. Then at the second level, class-specific algorithms allocate bandwidth within each class. A class-specific algorithm takes bandwidth allocated to the class by FLS and distributes it among active connections of the class. The proposed scheme is shown in Fig. 4.

FLS implements the priority of service classes by selecting an appropriate order of bandwidth allocation. Bandwidth is allocated to the service classes in the following order: UGS, ertPS, rtPS, nrtPS, and BE. In this way UGS has the highest priority, while BE class has the lowest priority. FLS allocates bandwidth such that following conditions are met:

1. QoS is ensured for all service classes
2. Higher priority flows are not be affected by lower priority flows
3. No service class starves
4. Efficient bandwidth utilization

The scheduling of UGS and ertPS classes are well-defined by the standard. Therefore, FLS distributes bandwidth among rtPS, nrtPS, and BE classes. FLS must guarantee bandwidth for nrtPS class, while both bandwidth and delay for rtPS class. FLS ensures that each class, except BE class, gets its MRTR. In each scheduling round, FLS first allocates $min(\sum_{i \epsilon F_o} r_i^{min}, \sum_{i \epsilon F_o} br_f^i)$ bandwidth to service class o, where $o \epsilon \{rtPS, nrtPS\}$. Henceforth, we will represent this allocated bandwidth by R_o. Since the MRTR for a BE connection is zero i.e. $r_i^{min} = 0$, therefore no bandwidth allocation could be done for BE class in this manner. Instead, a small portion of total uplink bandwidth, R_{BE}, is reserved for BE class. R_{BE} is not fixed and may vary for each MAC frame, however it is always less than or equal to $\sum_{i \epsilon F_{BE}} br_f^i$. The reserved bandwidth prevents the starvation of BE flows.

Since in each round R_{nrtPS} and R_{BE} are reserved for nrtPS and BE flows respectively. Therefore, $r^a - R_{nrtPS} - R_{BE}$ amount of bandwidth is available for rtPS class. This is the maximum amount of bandwidth that could be allocated to rtPS class. If the bandwidth utilized by rtPS class is less than $r^a - R_{nrtPS} - R_{BE}$, then the remaining bandwidth is allocated to nrtPS and BE classes. Thus the total bandwidth available to nrtPS connections is equal to R_{nrtPS} plus any underutilized bandwidth by rtPS class. After scheduling of rtPS and nrtPS traffic, the remaining bandwidth is allocated to BE connections. Obviously, at least R_{BE} bandwidth is always available for BE class.

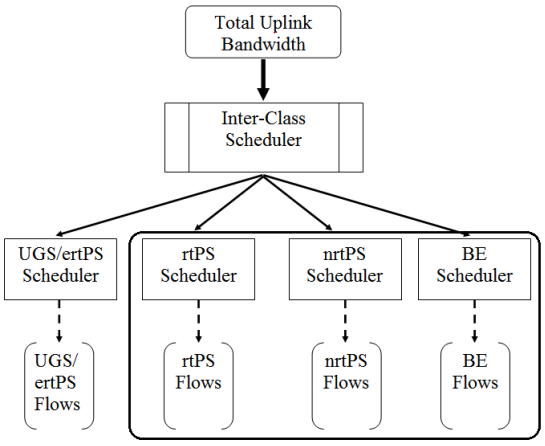


Fig. 4 The proposed Scheduling Scheme

4.4 Second Level Scheduling

4.4.1 rtPS Scheduling

Fairness In order to guarantee fairness among rtPS flows, we introduce a parameter called *Service Ratio*, which is computed for each connection as shown in Eq. 1. Another parameter, called *Total Service Ratio*, is also calculated as shown in Eq. 2. A connection i is only allowed to transmit data if $SR_i \leq SR$. If $SR_i > SR$, this signifies that there are some connections for which the *Service Ratio* is less than SR and therefore they should be given priority over connection i. The idea is to guarantee MRTR for each session, while fairly distributing the available bandwidth among active rtPS flows. It should be noted that in the ideal scenario all connections would have equal values of *Service Ratio*. Mathematically,
$SR_i = SR_j = SR$, where $i, j = 1, 2, 3, ..., n$

$$SR_i = \frac{\sum_{t=1}^{f-1} ba_i^t}{\sum_{t=1}^{f-1} br_i^t} \quad (1)$$

where, $i = 1, 2, ..., n$

$$SR = \frac{\sum_{t=1}^{f-1} \sum_{i=1}^{n} ba_i^t}{\sum_{t=1}^{f-1} \sum_{i=1}^{n} br_i^t} \quad (2)$$

Scheduling For each uplink frame the BS allocates bandwidth to connections in increasing order of d_i, i.e. priority is given to the connections with tighter delay constraints. In each frame, a connection i is allowed to transmit data only if $SR_i \leq SR$ or some bandwidth is available after scheduling other realtime connections. When a new bandwidth request br_i^f arrives, the BS tries to allocate the requested bandwidth to i in f. However if the available bandwidth to connection i in f is not sufficient to fulfill the request, then the scheduler tries to allocate the maximum possible bandwidth to i in f, while the remaining bandwidth is allocated to i in $f + d_i$.

To facilitate bandwidth allocation, the BS uplink scheduler maintains an $n \times d_{max}$ table, called bandwidth allocation table. In the rest of the article we would denote the bandwidth allocation table by symbol $bTbl$. An entry $bTbl[i][j]$ in $bTbl$ is an ordered pair (c, u), where c and u are bandwidth allocations to i, called $confirm\ allocation$ and $unconfirm\ allocation$ respectively. The allocation c is the guaranteed amount of bandwidth for i in frame j, while u is the bandwidth which could be allocated to i between frame f and $f+j$. However, there is no guarantee that the algorithm will actually make this allocation. The table is also used by the BS to generate UL-MAP. At the end of each scheduling round, the first column of $bTbl$ corresponds to the UL-MAP for the next uplink subframe. The generation of UL-MAP is explained in more details at the end of this section. The proposed scheduling algorithm for rtPS class is presented in algorithm 1. The step by step explanation of the proposed algorithm is provided in subsequent paragraphs.

The procedure schedule *schedule*, lines 1-13, is invoked at the start of each frame to schedule rtPS traffic. The for loop, lines 2-11, runs the scheduling algorithm for each connection's bandwidth request. Line 4 calls function $allocateBw$. The function tries to allocate the maximum possible bandwidth to i, to satisfy the bandwidth request br_i^f, in f. The function then returns the amount of bandwidth successfully allocated to the connection in f. So, this value is subtracted from the requested bandwidth to obtain the amount of bandwidth still to be allocated. The unallocated bandwidth could be scheduled between $f + 1$ and $f + d_i$. Instead of finding the exact column in $bTbl$ to make entry for this allocation, the algorithm tries to do the maximum possible allocation in frame $f + d_i$ (line 5). In fact, if some space would become available before $f + d_i$, the request would be scheduled earlier. If there is still some unallocated bandwidth, then it is assigned as $unconfirm\ allocation$ at $bTbl[i][f+d_i]$ (line 8). Later on, if some bandwidth would become available between frames $f + 1$ and $f + d_i$, this entry could be converted to $confirm\ allocation$. The condition at line 7 can be true either because $SR_i > SR$ and therefore no bandwidth allocation can be done for connection i or there is there is some bandwidth which could not be allocated in statements 4 and 5. Regardless of the case, an $unconfirm\ allocation$ is done at $bTbl[i][f + d_i]$ in line 8. Then, the algorithm updates SR_i and SR according to equations 1 and 2.

The function $allocateBw$ is invoked by the procedure *schedule* to make an entry in $bTbl$ for bandwidth allocation to a connection. The definition of $allocateBw$, lines 19-31, is self-explanatory and it is provided for the sake of completeness. The function then returns the amount of bandwidth that it is able to allocate for the connection in the specified frame.

The generation of UL-MAP is straightforward. At the end of procedure *schedule*, the first column of $bTbl$ corresponds to UL-MAP for the next uplink subframe. If some bandwidth is available in the next uplink subframe, then $confirm\ allocations$ with the earliest deadlines from subsequent frames are scheduled in the next frame. If no more $confirm\ allocations$ and there is still some bandwidth available then $unconfirm\ allocations$ can be scheduled in order of their deadlines. If a packet miss its deadline, it is proposed that the packet should be dropped by SS scheduler. The proposed algorithm is illustrated with the help of an example, which is presented at the end of this section.

The run-time complexity of the proposed algorithm is easy to calculate. Lines 3 to 10 are executed for each bandwidth request. Lines 4 and 5, call the function $allocateBw$. It can be seen that all steps in the functions are done in constant time. Therefore, the complexity of allocateBw is $O(1)$. Similarly, statements 7 to 10 are executed in constant time. Hence, for each bandwidth request, the run-time complexity of the proposed algorithm is $O(1)$.

As soon as a bandwidth request is received, the algorithm decides how much allocation to the connection could be done against this request. Therefore, there is no need for an SS to send the value of backlog as a bandwidth request, but it can actually send the amount of traffic generated between $f - 1$ and f. This approach has two distinct advantages: firstly less bandwidth is required to make requests, and secondly the BS need not to determine the deadline of individual packets through some complex procedure. However, even if an SS send the value of backlog as bandwidth request, we can use arrival-service curve [39] used in [19] to determine the actual new bandwidth demand generated during $f - 1$ and f. Mathematically, arrival-service can be represeted by equation 3, where $backlog(f)$ is the current

Algorithm 1 rtPS intra-class scheduling algorithm

```
1: procedure schedule()
2: for i = 1 to n do
3:    if SR_i ≤ SR then
4:       set br_i^f -= allocateBw(br_i^f, i, f)
5:       set br_c^f -= allocateBw(br_i^f, i, f + d_i)
6:    end if
7:    if br_i^f > 0 then
8:       set bTbl[i][f + d_i].u += br_i^f
9:    end if
10:   SR_i
11: end for
12: SR
13: generate UL-MAP
14: end procedure
15:
16:
17: {Function allocateBw attempts to reserve an amount bw
    of bandwidth for the connection conn in frame frame. It
    takes three parameters: (i) bw− bandwidth to allocate.
    (ii) conn− connection which request the bandwidth allo-
    cation. (iii) frame− frame in which the bandwidth is to
    be allocated.
    Returns the amount of bandwidth successfully allocated}
18:
19: function allocateBw(bw, conn, frame)
20: if bw ≤ 0 or r_frame ≤ 0 then
21:    return 0
22: end if
23: if r_frame ≥ bw then
24:    set allocate = bw
25: else
26:    set allocate = r_frame
27: end if
28: set r_frame -= allocate
29: set bTbl[conn][frame].c += allocate
30: return allocate
31: end function
```

bandwidth request, $backlog(f-1)$ is the previous bandwidth request, and $service(f-1)$ is the bandwidth allocated to the connection in the previous frame. However, in the case, expired packets are dropped by SS, we must add the packets dropped during period $[f-1, f]$ to equation 3. The corrected version is given in equation 4, where $drop(f-1, f)$ is the total bytes dropped during $f-1$ and f.

$$br_k^f = backlog(f) + service(f-1) - backlog(f-1) \quad (3)$$

$$br_k^f = backlog(f) + service(f-1, f) - backlog(f-1) + drop(f-1, f) \quad (4)$$

Illustrating Example We explain the working of the algorithm with the help of the example shown in figure 5. In this example, there are three connections to be scheduled: A,B and C with delay limits of 30ms, 40ms and 60ms respectively. We assume total uplink bandwidth per frame to be 10 units and a frame duration of 20ms. This implies that a packet generated by A,B and C between $f-1$ and f must be scheduled within next 1,2 and 3 frames respectively. The bandwidth requests generated by the three connections are shown in column 2. For example, the first entry in the first row of column 2, is the amount of traffic that arrived in the input queue of connection A for uplink transmission between frame 0 and frame 1. The bandwidth request for this traffic will be treated at the start of frame 1 by the BS uplink scheduler. The third column shows the values of SR and SR_i at the start of scheduling f. An entry in the fourth column is the *bTbl* that is obtained at the end of scheduling frame f. The shaded entries in *bTbl* are *unconfirmed allocations*. The underline entries in *BTbl* are the allocations done during current frame. UL-MAP corresponding to next uplink subframe is shown in the fifth column.

The scheduling in the example is done as follows. The algorithm is able to schedule the requested bandwidths in [0,1]. Note specially the allocations done for connections B and C. Since only 10 units can be allocated in a frame, therefore we cannot do *confirmed-allocation* of more than 10 units within in column. For scheduling the requests in [1,2], $SR_A \leq SR$ but no bandwidth is available in the current frame. Furthermore, due to delay limits this request cannot fulfilled in subsequent frames. Therefore, it is entered as an *unconfirmed allocation* in the column corresponding to next frame. As there is no provision in the current frame, therefore this request is not scheduled in UL-MAP of frame 2. In the duration [2,3], B request 12 units of bandwidth. Since $SR_B > SR$, therefore the algorithm allocates it as an *unconfirmed allocation* in the frame $f + d_B$ i.e. in frame 4. The unused 5 units of bandwidth in frame 3 are used to schedule 5 units of bandwidth from the next frame. For the duration [3,4], there is no bandwidth request. There is a *confirmed allocation* of 5 units and 2 *unconfirmed-allocation* of 17 units. Therefore, 5 units can be allocated to first *unconfirmed allocation* for B. The remaining bandwidth demand cannot be fulfilled. The final values of SR_i and SR are shown in the last row.

It is important to understand that all unexpired packets belonging to the same connection are always scheduled in the order of their deadlines by SS scheduler. The important thing is the amount of bandwidth allocated to the connection and not the actual packets against which the allocations are done. This is due to the fact that SS scheduler transmits packets in first-in first-out (FIFO) order. Consider the example given in figure 6. We assume two connections A and B, with $d_A = d_B = 2$. Note that the BS grants 5 units to A against demand of 10 units. However, the SS sched-

uler schedules the packets which are at the front of A's queue. Note, however, the 5 units were granted against the second packet in the queue and not the packet at the front of the queue.

4.4.2 nrtPS Scheduling

The nrtPS scheduling is done in two stages. Firstly, the algorithm makes sure that every connection gets at least its MRTR. In the second stage, the algorithm allocates more bandwidth to connections with greater backlog. Let for $u \epsilon F_{nrtPS}$, r_u^{cur} be the current bandwidth demand. Then, $\forall u$, the algorithm first allocates $min(r_u^{cur}, r_u^{min})$ amount of bandwidth to u. Let $bLog_u$ be the backlog of u after allocation in the first stage and r_{avl} be the amount of bandwidth still available in f for nrtPS flows. In the second stage, r_{avl} is distributed among nrtPS connections in proportion of their backlogs. Mathematically, the total bandwidth assigned to u is shown in Eq. 5:

$$min\left(r_u^{cur}, r_u^{min}\right) + min\left(r_{avl}, \sum_{u \epsilon F_{nrtPS}} bLog_u\right) \times \left(\frac{bLog_u}{\sum_{u \epsilon F_{nrtPS}} bLog_u}\right) \quad (5)$$

The scheme ensures that each nrtPS connection gets at least the MRTR. Furthermore, using $bLog_u$ as weight enables the algorithm to accelerate data transmission for more demanding connections. Therefore it can be considered as a need-based scheme, which also ensures QoS for all nrtPS connections. The scheme guarantees that a high-bandwidth source cannot inundate network resources.

We illustrate the proposed scheme with the help of an example. Lets consider there are three nrtPS connections to be scheduled with parameters as shown in Table 3. For simplicity, we assume that the length of each frame is 1 second. Therefore, each connection send minimum 3 units of data in a second. We further assume that the total available bandwidth is 12 units and the actual data generation rate is 3,6 and 9 units per second for connections n1, n2, and n3 respectively. The bandwidth allocation by the proposed nrtPS scheduling algorithm and the backlog after each allocation is shown in Table 4. The bandwidth requested by a connection in each frame is equal to its current backlog i.e. backlog in the last frame plus the bandwidth required for newly arrived packets. The algorithm first allocates

Connection	MRTR	MSTR	Actual Rate
n1	3	5	3
n2	3	10	6
n3	3	15	9

Table 3 Parameters of connections for nrtPS scheduling example

Connection		Frames			
		f_1	f_2	f_3	f_4
n1	$\frac{allocation}{request}$	$\frac{3}{3}$	$\frac{3}{3}$	$\frac{3}{3}$	$\frac{3}{3}$
	backlog	0	0	0	0
n2	$\frac{allocation}{request}$	$\frac{4}{6}$	$\frac{4}{8}$	$\frac{4}{10}$	$\frac{4}{12}$
	backlog	2	4	6	8
n3	$\frac{allocation}{request}$	$\frac{5}{9}$	$\frac{5}{13}$	$\frac{5}{17}$	$\frac{5}{21}$
	backlog	4	8	12	16

Table 4 Bandwidth allocation for nrtPS scheduling example

3 units (MRTR) of bandwidth to each connection, then the remaining 3 units of bandwidth is distributed according to backlog as explained in Equation 5.

4.4.3 BE Scheduling

The allocation of bandwidth at physical layer is done in terms of number of time slots. An SS with bad channel conditions consume more time slots for transmitting relatively small amount of data. On the other hand, an SS with good channel conditions can send much more data in the same number of time slots. Therefore, we propose to distribute the available time slots equally among BE connections so as to maximize the usage of bandwidth. Let C be the number of available time slots for BE traffic, and n_{be} be the number of BE connections. Then the number of slots available per connection can be given as C/n_{be}. For a BE connection v, let r_v^{cur} be the current bandwidth request and C_v time slots are required to fulfill the request. Then the algorithm allocates $min(C_v, C/n_{be})$ time slots to v. An SS with good channel conditions will be able to send more data within same number of time slots than an SS with poor channel conditions and thus automatically get prioritized. This scheme thus prevents SS with poor channel conditions to affect other SSs.

The difference between equal bandwidth allocation and equal time slot allocation can be illustrated with the help of an example. Suppose there are four SS: $S1$, $S2$, $S3$ and $S4$ with one BE connection each. Let the first three SSs have good channel conditions and in each time slot they can send 5 units of data , while $S4$ has poor channel conditions and it can send only 1 unit of data per slot. We also assume that 16 time slots are available for BE traffic. Then the bandwidth allocation under the two schemes is shown in Fig. 7. Under equal

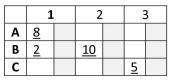
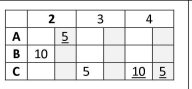
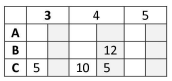
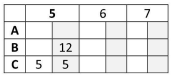
Duration	Bandwidth Requests			Service Ratios				Bandwidth Allocation Table	UL-MAP		
[f-1,f]	A	B	C	SR_A	SR_B	SR_C	SR		A	B	C
[0,1]	8	12	5	1.00	1.00	1.00	1.00		8	2	0
[1,2]	5	0	15	1.00	1.00	1.00	1.00		0	10	0
[2,3]	0	12	0	0.62	1.00	0.75	0.78		0	0	10
[3,4]	0	0	0	0.62	0.50	0.75	0.61		0	5	5
				0.62	0.71	0.75	0.70				

Fig. 5 An example of scheduling several frames to demonstrate the working of rtPS class algorithm

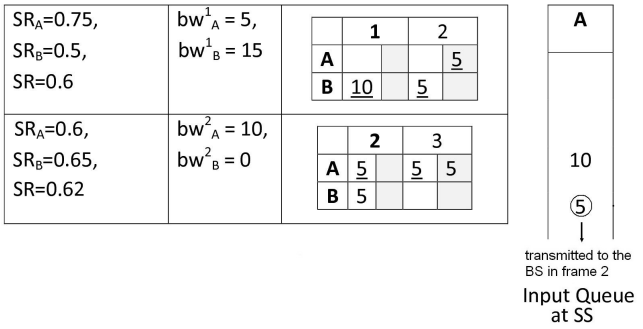


Fig. 6 FIFO scheduling at SS

bandwidth distribution $S4$ is able to reduce the bandwidth of other connections by 50%. There is no QoS to guarantee and $S1$, $S2$ and $S3$ have good channel conditions but still they are paying the penalty of poor channel conditions of $S4$. Clearly, equal slot allocation makes use of bandwidth much more efficiently.

5 Video Transmission Mechanism

One of the main applications of rtPS class is the transmission of realtime videos. Therefore we designed and implemented a realtime video transmission mechanism (VTM) and implemented it in Qualnet. The aim is to assess the performance of the proposed rtPS scheduler in scheduling realtime video streams.

VTM can send videos encoded in High Efficiency Video Coding (HEVC), also known as H.265 and MPEG-H Part 2 [43]. It is a block-based hybrid video coding standard that provides inter/intra prediction and transform coding with high efficiency entropy encoding [44]. It is a successor to widely used H.264/MPEG-4 Advanced Video Coding (AVC) standard. For same level of video quality, it offers double the data compression ratio as compared to AVC. Consequently, HEVC provides much better quality of video at the same data rate [43]. HEVC provides many enhancements over H.264/AVC:

(a)

	S1	S2	S3	S4	Total
Bandwidth Allocated	10	10	10	10	40
Time Slots Required	2	2	2	10	16

(b)

	S1	S2	S3	S4	Total
Time Slots Allocated	4	4	4	4	16
Available Bandwidth	20	20	20	4	64

Fig. 7 (a) Equal bandwidth distribution (b) Equal time slot distribution

- 64x64 pattern comparison and difference-coding
- Improved variable-block-size segmentation
- Better prediction within the same frame
- Better motion vector prediction
- Sample-adaptive offset filtering
- Improved motion region merging

HEVC was developed by a collaboration between ISO/IEC MPEG and ITU-T VCEG. The first version of the standard was issued in June 2013. The second version, published in 2015, introduced multi-view extension (MV-HEVC), range extension, and scalability extension (SHVC). Extensions for 3D video (3D-HEVC) and screen content coding (SCC) were published in 2016 and 2017, respectively. A brief history of HEVC versions is presented in Table 5.

Version No.	Year	Key Features
1	2013	First approved version containing Main, Main10, and Main Still Picture profiles.
2	2014	Adds a multi-view extension profile, scalable extension profiles, and 21 range extension profile.
3	2015	3D Main profile
4	2016	Adds four scalable extension profiles, three high throughput extension profiles, and additional screen content coding extensions profiles.

Table 5 A brief history of HEVC [45]

Standard	H.264 HP	MPEG-4 AP	H.263 HLP	H.262 MP
Bitrate Reduction	35.4%	63.7%	65.1%	70.8%

Table 6 Bitrate Reduction Offered by HEVC against different video encoding standards

An HEVC profile is a set of features and associated coding tools that allows to create a video stream that conforms to the features specified in that profile. There are three profiles in Version 1 of HEVC: Main, Main10, and Main Still. Version 2 added several new profiles that provide extensions such as increased bit-depth, multi-view video coding, screen content coding, and scalable video coding.

The aim of HEVC is to provide high coding efficiency by minimizing bit rate while still maintaining a certain level of video quality. Coding efficiency can be measured either by using objective metrics such as peak signal-to-noise ratio (PNSR), or subjective assessment. Subjective tests, such as mean opinion score, involves assessment of video quality by several individuals and it is the generally the preferred way.

Ohm J-R et al. [46], compared the efficiency of HEVC Main profiles against H.264 High Profile (HP), MPEG-4 Advanced Profile (AP), H.263 High Latency Profile (HLP), and H.262 Main Profile. Nine test sequences were encoded at twelve different bitrates by using HM-8.0 HEVC encoder. Peak Signal-to-Noise ratio (PNSR) was used as an evaluation criterion. The bit-rate reduction offered by HEVC standard is summarized in Table 6.

In another study [47], subjective tests were conducted to evaluate the quality of HEVC and H.264 / MPEG-4 AVC HP. Three 5 second video sequences of resolutions 3840x1744 at 24 fps, 3840x2048 at 30 fps, and 3840x2160 at 30 fps were encoded at five different bitrates with HM-6.1.1 HEVC encoder and the JM-18.3 H.264/MPEG-4 AVC encoder. The results revealed that HEVC provided an average bitrate reduction of 66.5%, based on mean opinion score, as compared to H.264/MPEG-4 AVC HP.

HEVC works by comparing different parts of a video frame to find areas that are similar or redundant, both within a single video frame as well as subsequent frames. These areas are then replaced with a compressed code that requires significantly less bits than original pixels.

The basic processing unit in HEVC is coding tree unit (CTU), also known as largest coding unit. It is conceptually similar to macroblock units that were used in several earlier generation standards. It has been empirically shown that larger CTU sizes increases coding efficiency while at the same time reduces decoding time [46].

The HEVC video coding layer uses a hybrid approach utilizing both inter-frame and intra-frame prediction and 2D transform coding. In intra-frame prediction, the prediction of regions in a frame is based only on the information from the same frame. While inter-frame prediction requires information from other frames. The final frame representation after prediction is stored a decoded frame buffer that can then be used for subsequent predictions.

An HEVC bitstream is organized into network abstraction layer (NAL) that makes video layer to be transparent to various transmission mechanisms. NAL units are of two types, i.e. video coding layer (VCL) and non-video coding layer (non-VCL) units. VCL units contain data associated with coded video, while non-VCL contains data shared by different video frames. A NAL unit consists of a fixed two-byte header and associated payload. Each NAL unit has a unique TemporalID that specifies the temporal layer associated with that unit. All NAL units of a given video frame have same TemporalID. Therefore, each video frame has a unique value of TemporalID.

For the experiments we used the reference software for HEVC called HEVC Test Model (HM) [44]. The software provides a reference implementation of the HEVC standard which is developed by the Joint Collaborative Team on Video Coding (JCT-VC) that aims to provide a basis upon which to conduct experiments involving HEVC standard [48]. We specifically used HM-16.0 encoder. The working of HM encoder is shown in Figure 8 and briefly discussed in subsequent paragraphs.

The input video is first divided into coding tree units that are further split using a quadtree into coding units (CU). The leaf CU defines the shape of prediction units and a residual-quadtree containing transformation units. A transformation unit defines a region containing the same transformation and quantization process.

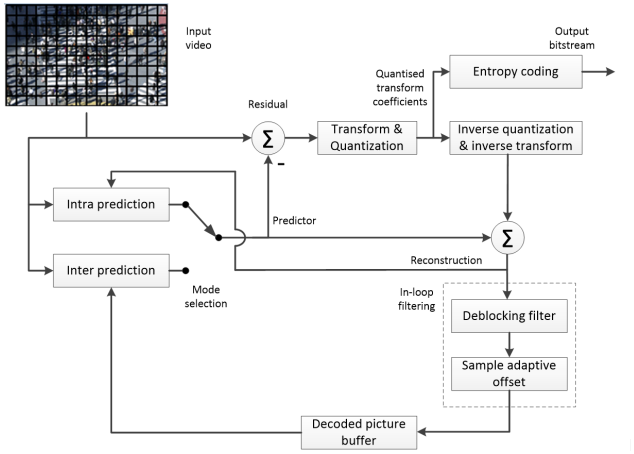


Fig. 8 Simplified Block Diagram of HM Encoder [44]

The intra prediction block provides 35 modes for the luma of each prediction unit. Prediction efficiency is improved by applying mode-dependent reference and sample smoothing. The intra prediction mode is then coded by using one of the three most probable modes. The inter prediction involves selecting motion parameters such as skip mode, merge mode, and motion vectors. The transform and quantization block takes the prediction from the input and apply spatial transformation and quantization on it. Entropy coding is then applied to the generated symbols and quantized transform coefficients using a Context-based Adaptive Binary Arithmetic Coding [49].

Deblocking filtering and sample adaptive offset filtering are two in-loop filtering processes. The processes are applied after the reconstructed pixel data is formed. The image is the stored in decoded picture buffer that may be used for predicting content of subsequent frames.

To perform the experiment we used the video sequences obtained from the video trace library of Arizona State University [50]. The video sequences are in raw .YUV format. Firstly, a raw video file is encoded in HEVC bitstream by HM-16.0 encoder. Then a packet trace file for the encoded video is generated. A packet trace specifies the size of packets and their parameter values. A sample packet trace file is shown in 9. The packet trace file and the encoded bitstream are read by the VTM. VTM then transmits the encoded bitstream, by using TLSA, according to the parameters specified in the packet trace. The transmitted video is received at the receiver and a corresponding HEVC bitstream is generated. The bitstream is then decoded to obtain the corresponding distorted raw YUV file. Finally, the quality of the received video can be compared with the original video transmitted. The process is depicted in Fig 10.

Start-Pos.	Length	LId	TId	QId	Packet-Type	Discardable	Truncatable
0x00000000	262	0	0	0	StreamHeader	No	No
0x00000106	14	0	0	0	ParameterSet	No	No
0x00000114	16	0	0	0	ParameterSet	No	No
0x00006fd4	238	1	0	0	SliceData	No	No
0x000070c2	9	0	1	0	SliceData	Yes	No
0x000070cb	979	0	0	0	SliceData	No	No
0x0000749e	9	0	1	0	SliceData	Yes	No
0x000074a7	121	0	0	0	SliceData	No	No
0x00007520	976	1	1	0	SliceData	Yes	No
0x000078f0	598	1	1	0	SliceData	Yes	No
0x00007b46	9	0	2	0	SliceData	Yes	No
0x00007b4f	679	0	0	0	SliceData	No	No
0x00007df6	979	1	2	0	SliceData	Yes	No
0x000081c9	9	0	2	0	SliceData	Yes	No
0x000081d2	768	0	0	0	SliceData	No	No
0x000084d2	983	1	2	0	SliceData	Yes	No
0x000088a9	97	1	2	0	SliceData	Yes	No

Fig. 9 A sample trace file

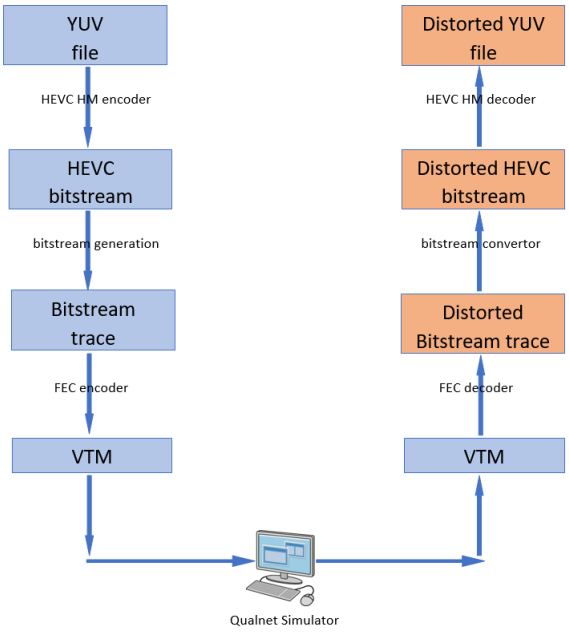


Fig. 10 Framework for realtime video transmission

6 Simulation Results

The performance of TLSA is evaluated by developing a simulation model. Qualnet v5.02 [51] is used to perform all simulations. Qualnet is a commercial network simulator that provides good support for the IEEE 802.16 standard. The implementation of 802.16 standard in Qualnet is done in *Advanced Wireless Model* library [52]. The library provides an extensive set of customizable parameters and a faithful implementation of the 802.16 standard. The source code of the library is available and the developers can modify the code to implement new algorithms and protocols.

For each simulation, data transmission is started at 20s. This delay is necessary for proper functioning of routing protocols. Since the actual transmission starts at 20s, so we consider this as the starting point of simulation i.e. $t = 0s$. We repeat each simulation 50 times and then average results are presented in this section. For each repetition, a different seed is used to alter the

Parameter	Value
Total uplink bandwidth	1 Mbps
Frame duration	20 ms
MAC propagation delay	1 μs
Cyclic prefix	8.0
Antenna model	omni antenna
Sampling factor	144/125
Propagation model	Two ray ground
Timeout interval	15 s
Antenna height	1.5 m
Antenna gain	1
Transmit power	20 dBm
Receive power threshold	205e-12
Carrier sense power threshold	0.9 * Receive power threshold
Link adaptation	Enabled

Table 7 Important simulation parameters

characteristics of simulation such as patterns of traffic generation and mobility. It is assumed that packets only arrive at start of a frame and all connections are admitted. The values of important parameters used for simulation are presented in table 7.

6.1 FLS Algorithm

6.1.1 Bandwidth Allocation among Fixed Subscriber Stations

The purpose of this experiment is to assess the performance of FLS algorithm. For the simulation, BE traffic is generated at an average rate of 200 Kbps throughout the experiment. Approximately 100 Kbps of bandwidth is reserved for BE traffic to prevent it from starvation. While for nrtPS the MRTR is 400 Kbps and the average traffic rate is 580 Kbps. Simulations are performed with increasing load of rtPS traffic. Initially, the average traffic rate of rtPS is 300 Kbps, which is gradually increased to 600 Kbps. The MRTR for rtPS traffic is 300 Kbps throughout the experiment, while the maximum allowed delay is set to 160ms.

The bandwidth distribution by FLS is shown in Fig. 11. As the data generation rate of rtPS class is increased from 300 Kbps to 400 Kbps, the throughput of BE traffic is reduced from 200 Kbps to 100 Kbps. While there is no effect on the throughput of nrtPS traffic. As the rtPS data rate is further increased, the throughput of nrtPS decreases. Since 100 Kbps is the reserved bandwidth for BE class, therefore the throughput of BE traffic cannot be further reduced and remains unaffected. When rtPS data generation rate is increased to 520 Kbps, the throughput of nrtPS reaches its MRTR

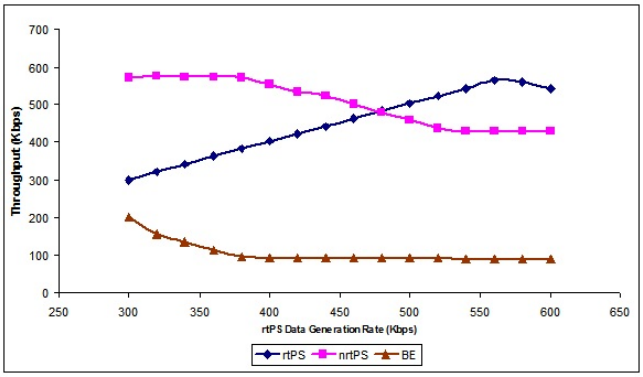


Fig. 11 Bandwidth distribution by FLS among fixed SSs

i.e. 400 Kbps. Further increase in rtPS traffic rate has no effect on the throughput of nrtPS and BE classes. So the throughput of rtPS cannot be further increased by just increasing its traffic generation rate. It can be seen that FLS is able to ensure that rtPS and nrtPS classes get at least their MRTR. In case of overload rtPS gets the priority and FLS takes away extra bandwidth from nrtPS and BE classes.

The percentage of lost packets is shown in Fig. 12. The percentage of lost packets is negligible till the total data generation rate is less than the available uplink bandwidth. Packet loss starts, once the data generation rate exceeds the available bandwidth. This is due to the fact that the packets that miss their deadlines due to overload are dropped at SSs.

Simulations are also performed to determine if FLS is able to meet the deadlines of rtPS traffic. The end-to-end delay observed by different service classes is shown in Fig 13. It can be seen that rtPS traffic observed the least delay. In fact, the end-to-end delay of rtPS traffic remains around 30 ms throughout the experiment, while the maximum allowed delay is 160 ms. Increase in rtPS throughput results in reduced bandwidth allocation to nrtPS and BE classes, which in turn results in relatively higher delays for these classes.

6.1.2 Effects of Mobility on FLS

Mobility of SSs can adversely affect the QoS provided by the network. This experiment is designed to determine the effect of mobility on the performance of FLS. Two BSs and three SSs are used in the simulation. Each SS has one connection of each service class (rtPS, nrtPS and BE). The SSs move linearly at a constant speed of 16.67 m/s and performs one handover during the simulation. Initially only BE traffic is present. The traffic rate of BE traffic is gradually increased from 20 Kbps to 160 Kbps (0-40 sec). After 40 second, the average rate

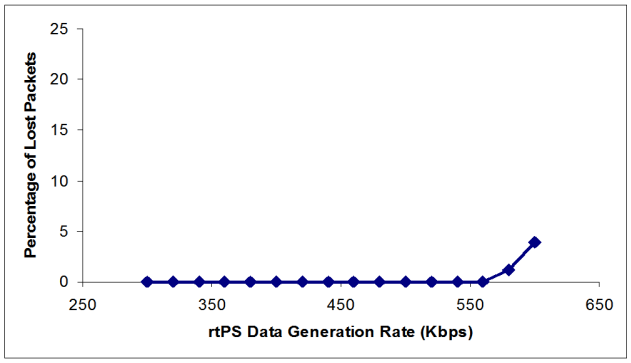


Fig. 12 Lost packets

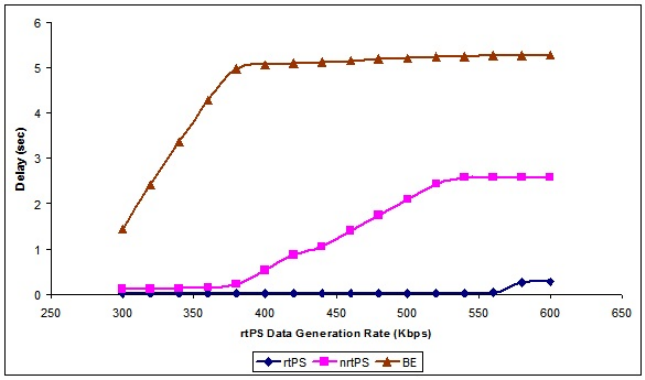


Fig. 13 End-to-end delay for different service classes under *FLS*

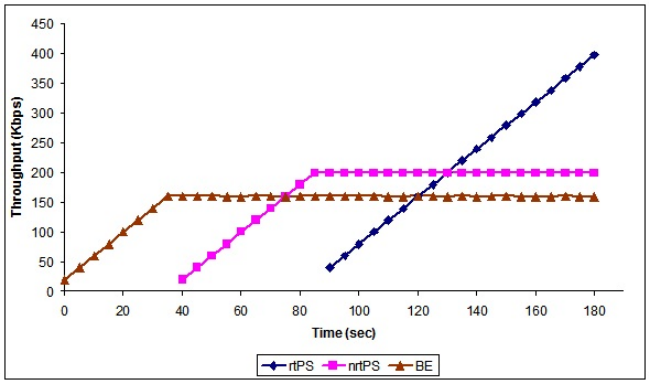


Fig. 14 Throughput of different classes of traffic for mobile SS

of BE traffic is kept constant. At 40th second nrtPS traffic is introduced in the network. The rate of nrtPS traffic is gradually increased to 200 Kbps (40-85 sec). After this point, the average traffic rate of nrtPS is kept constant at 200 Kbps. rtPS traffic is introduced at this point and its data generation rate is increased gradually to 400 Kbps (85-180 sec).

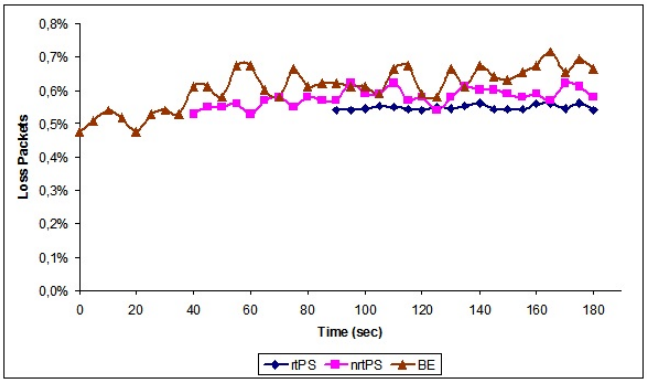


Fig. 15 The percentage of lost packets for mobile SS

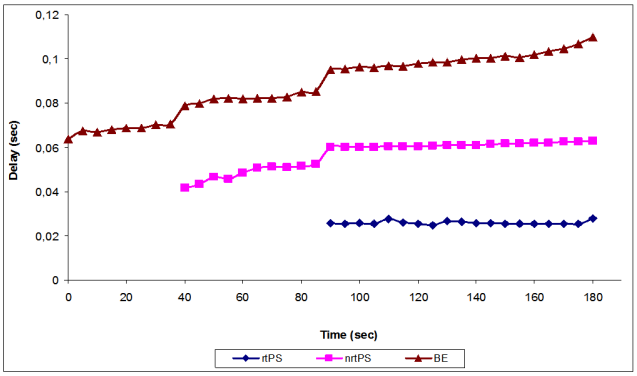


Fig. 16 Delay in mixed traffic network under mobility

The throughput of all service classes at the receiver is shown in Fig. 14. As the applied load is less than the available bandwidth, *FLS* is able to allocate bandwidth to service classes that exactly matches the input traffic pattern.

The percentage of lost packets is shown in Fig. 15. It can be seen that the percentage of lost packets remain below 0.75% for all classes of traffic. Furthermore, the fluctuation is the least in case of rtPS traffic. The percentage of lost packets is minimum for rtPS traffic, while maximum for BE traffic. However, the difference is not more than 0.1%. It can be seen that under normal load, the introduction of nrtPS traffic and rtPS traffic does not have significant effect on BE traffic.

The end-to-end delay for different classes of traffic is shown in Fig. 16. The introduction of nrtPS increases delay for BE traffic. Similarly, the introduction of rtPS traffic results in slight increase in delays for nrtPS and BE classes. The delay of rtPS traffic remains constant irrespective of applied load and is around 25 ms, which is very good for realtime traffic. Also, the delay of nrtPS traffic remains below 70 ms throughout the experiment.

Connection	MRTR (Kbps)	Average Traffic Rate (Kbps)
N1	140	200
N2	200	225
N3	225	275
N4	250	300
Total	815	1000

Table 8 Input Traffic Parameters for nrtPS Connections

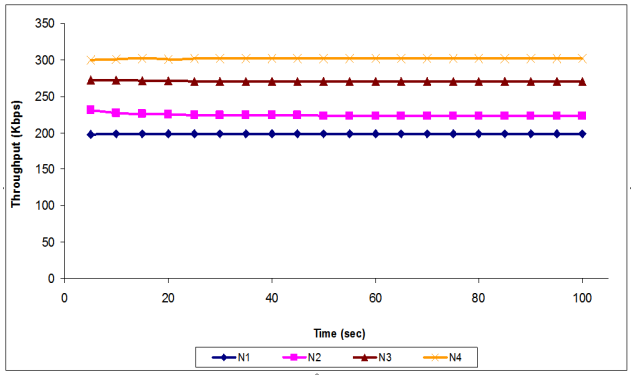


Fig. 17 Bandwidth allocation by nrtPS class specific algorithm

6.2 Second Level Scheduling

6.2.1 nrtPS Intra-class Scheduling

The experiment is performed to analyze the performance of nrtPS intra-class scheduling algorithm under high load i.e. the data generation rate is almost equal to the total available uplink bandwidth. Four SSs with one nrtPS connection each are used in the simulation. The parameters of the connections as shown in table 8. Note that the only type of traffic present is nrtPS and the ratio of available bandwidth to the applied load is almost 1.

The corresponding bandwidth allocation is shown in Fig. 17. It can be seen that throughput remains at a stable level for all connections. Furthermore, the MRTR is guaranteed for all nrtPS connections. We also calculated the service ratio (SR) as defined in equation 1. SR for N1, N2, N3 and N4 are approximately 0.99, 0.99, 0.98 and 0.98 respectively.

The end-to-end delays experienced by the four connections are shown in Fig. 18. Since all the connections get approximately same service ratio, therefore the end-to-end delay is identical for all connections. The maximum delay is observed by $N4$, which is slightly less than 0.5 seconds.

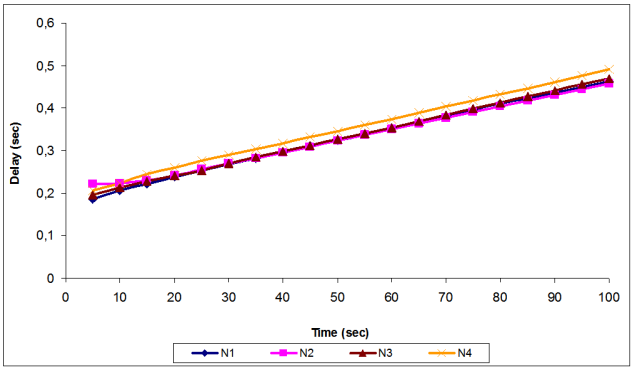


Fig. 18 End-to-end delay for nrtPS connections

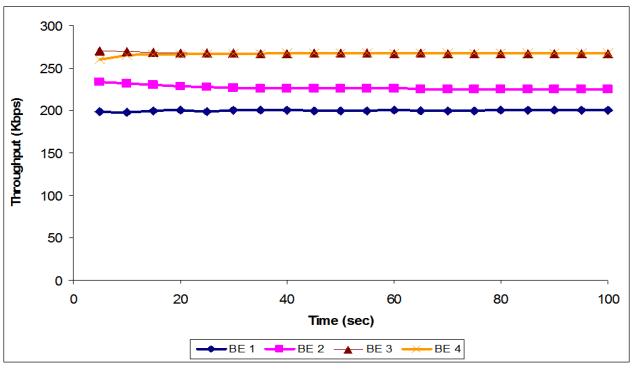


Fig. 19 Bandwidth allocation by BE class specific algorithm

6.2.2 BE Intra-Class Scheduling

To analyze the bandwidth allocation among BE connections, four BE connections $BE1$, $BE2$, $BE3$ and $BE4$ are used. Each connection is managed by a unique SS. The average data generation rate is 200 Kbps, 225 Kbps, 275 Kbps, and 300 Kbps for $BE1$, $BE2$, $BE3$ and $BE4$ respectively. Again, the ratio of available bandwidth to applied load is almost 1 and only BE traffic is used for the experiment.

The throughput achieved by the connections is shown in Fig. 19. The algorithm equally divides the available time slots among active BE connections. However, the data generation rate of $BE1$ and $BE2$ is less than the available bandwidth per connection. Therefore, the throughputs of $BE1$ and $BE2$ are equal to their data generation rates. The remaining bandwidth is distributed among other two connections.

The end-to-end delay experienced by the connections is shown in Fig. 20. The delay is almost negligible for $BE1$ and $BE2$, while it has the greatest value for $BE4$. Since for $BE1$ and $BE2$, the allocated bandwidth is equal to their data generation rate, therefore the input queues remain almost empty and thus the waiting

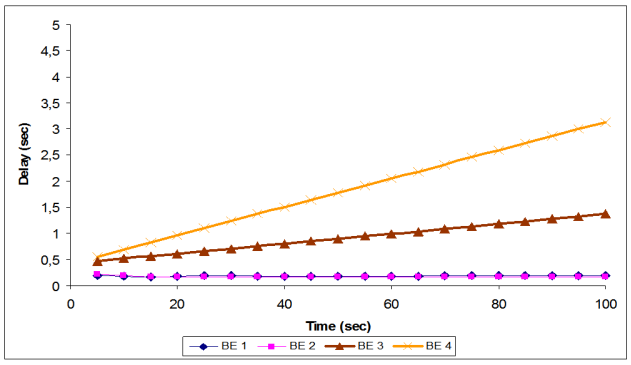


Fig. 20 End-to-end delay for BE traffic

time in the queue is negligible. While, the difference of throughput and data generation rate is maximum for $BE4$. Therefore, more and more packets wait in the input queue with the passage of time and thus the connection has relatively large delays.

6.2.3 rtPS Intra-Class Scheduling

Performance Analysis The objective of this experiment is to perform detailed analysis of rtPS intra-class scheduling algorithm. We also performed the same simulation for EDF algorithm and provide comparative results. For this experiment, the total uplink bandwidth is set to 10 Mbps. Four fixed SS with one rtPS connection each is used in the experiment. The parameters of the connections are shown in table 9. These parameters imply a very heavy load on system as the ratio of available bandwidth to data generation rate is less than 0.5.

Conn	MRTR (Kbps)	MSTR (Kbps)	Tolerable delay (frames)
A	4000	9000	2
B	1000	3000	3
C	2000	4000	3
D	3000	5000	4

Table 9 Input traffic parameters for rtPS connections

The service ratio for each connection and the total service ratio (SR) obtained during the simulation are shown in Fig. 21. It can be seen that service ratios of all rtPS connections adapt and follow SR. Even though the available bandwidth could only provide minimum guaranteed service to each connection, the proposed algorithm performed very well and dynamically allocate bandwidth to ensure fairness. In fact, SR is the best a connection can get and all the connections seem to follow SR rather well. Thus it shows that the algorithm is able to fairly allocate maximum possible bandwidth to each admitted rtPS connection.

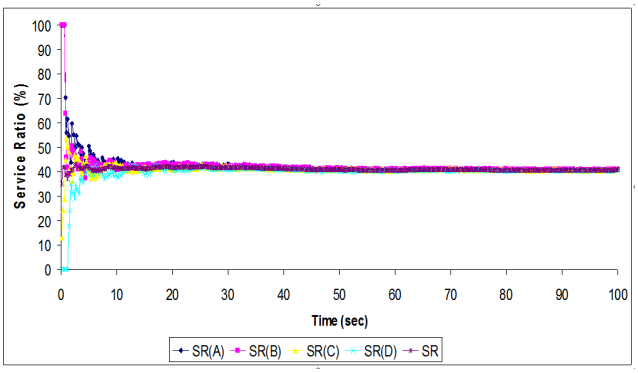


Fig. 21 Service ratio for rtPS connections by applying the proposed algorithm

Fig. 22 shows the service ratios obtained by scheduling using EDF on the same set of connections. There is not much difference between SR provided by EDF to that of provided by our proposed algorithm. However, obviously there is greater difference among the SR_i for individual connections. In this case, EDF allocates maximum bandwidth to A, while least bandwidth is allocated to B. This dispersion in *service ratios* is due to the fact that EDF tries to minimize the average delay but does not take fairness into account.

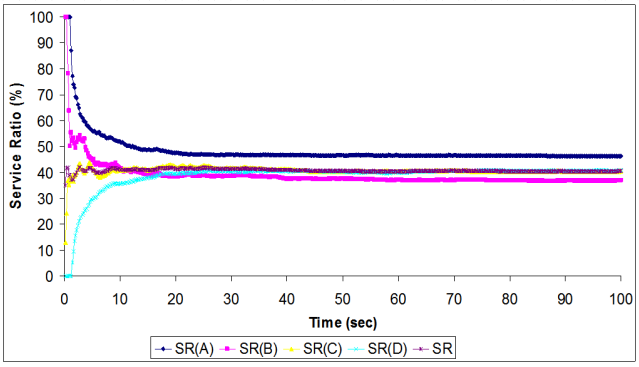


Fig. 22 Service ratio for rtPS connections by applying EDF

Fig. 23 shows the throughput as function of load. Clearly both algorithm are able to schedule all traffic until the load surpasses the available bandwidth of 10 Mbps. After this point, no matter how much load is applied the algorithms cannot give more throughput. However, EDF tends to drop some packets and throughput is slightly less than 10 Mbps.

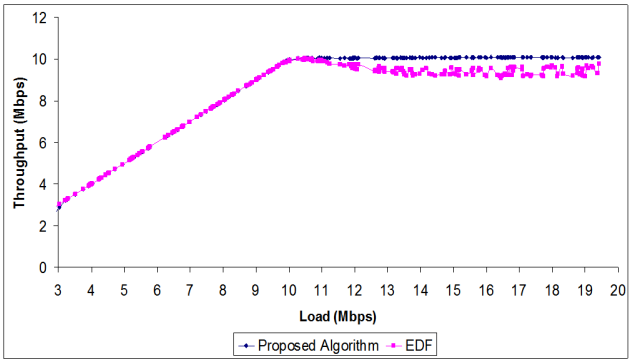


Fig. 23 Throughput vs applied load

Fig. 24 represents average delay packets experience as function of applied load. Under light and medium load conditions the packets are scheduled almost immediately by both algorithms. However, under very heavy load the packets have to wait more than the average waiting time. Note that expired packets are automatically dropped by SS. This effect is evident in the graph.

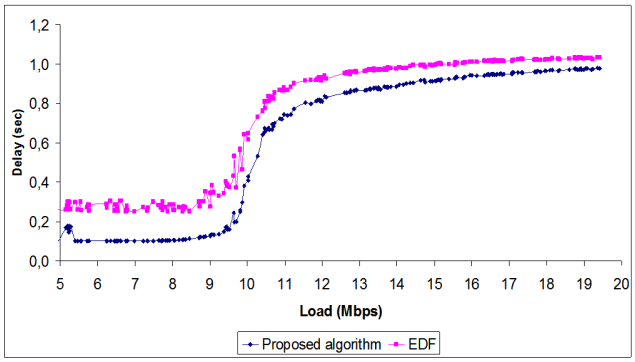


Fig. 24 Average Delay vs Load

Fig. 25 shows the ratio of loss packets as a function of load. When the applied load is less than 10 Mbps, both algorithms are able to schedule almost all the input packets and therefore the *loss packet ratio* is almost 0. Any traffic above 10 Mbps threshold cannot be scheduled and therefore the *loss packet ratio* increases sharply after this point. It can be seen that at a load of 20 Mbps, half of the traffic is dropped and so the *loss packet ratio* is around 0.5.

Lost packets as function of load under mobility The objective of this experiment is to determine the percentage of lost packets for rtPS class as function of load with mobile SS. The results of the experiment are presented in Fig. 26. For this simulation, the speed of SS is set to 60 Km/h (16.67 m/s) and it performs one handover.

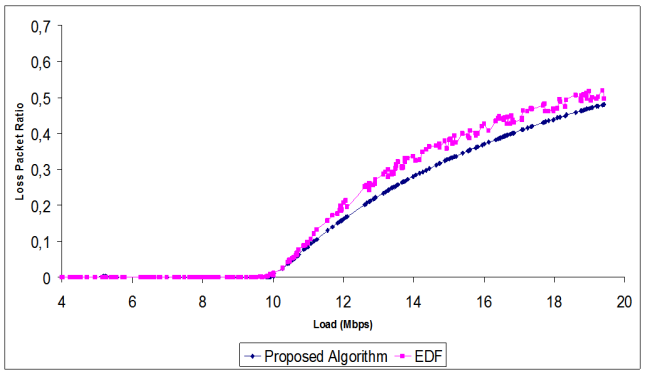


Fig. 25 Loss Packet Ratio vs Load

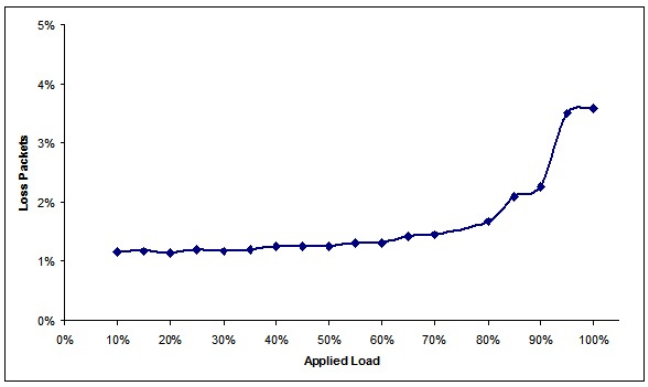


Fig. 26 Lost packets as function of traffic load under mobility

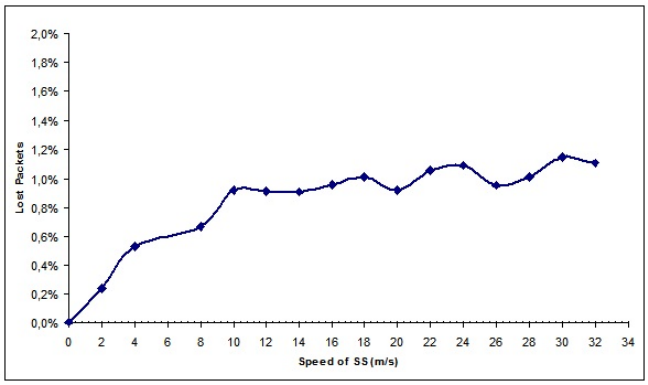


Fig. 27 Lost packets as function of SS speed

Simulations are performed with increasingly more load till the rtPS data generation rate is equal to total available uplink bandwidth. It can be seen that there is little increase in lost packets till the applied load is 80% of the available bandwidth. Further increase in load results in greater percentage of lost packets. However, the percentage always remain below 4%.

Lost packets as function of SS speed for rtPS class The experiment is performed to determine the percentage

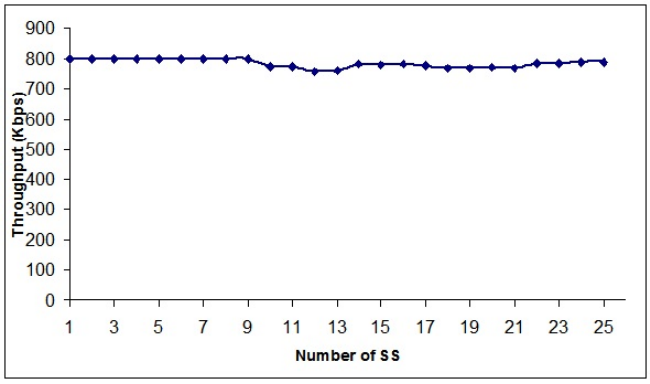


Fig. 28 Scalability of rtPS class specific algorithm

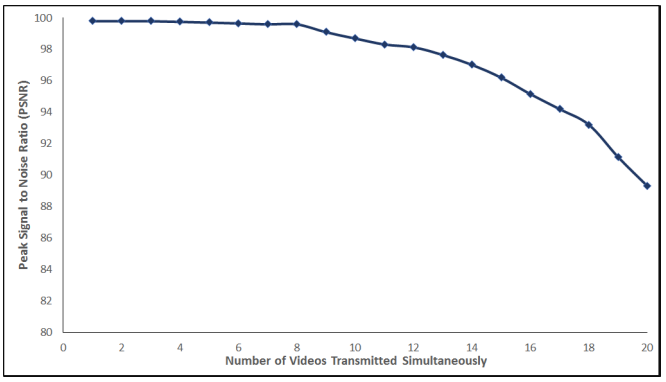


Fig. 29 Video Quality at the Receiver

of lost packets for rtPS class as function of SS speed. Three BS and one SS is used in the experiment. The SS traverses a distance of 10 km and perform two handovers. The SS has an rtPS connection with average data generation rate of 200 Kbps. Fig. 27 shows the effect of SS speed on uplink transmission. It can be seen that there are no lost packets when the SS is stationary. The percentage of lost packets increases relatively quickly between 0m/s and 10m/s. Further increase in SS speed, result in less significant increase in lost packets. The percentage of lost packets always remain below 1.2%. It should be noted that, in this simulation, the lost of some packets is due to physical layer phenomena and not because of the scheduling algorithm.

Scalability Scalability of a scheduling mechanism is a highly desirable property, especially for the scheduling of delay sensitive traffic such as rtPS. Therefore, this experiment is performed to determine the effect of number of SSs on the performance of rtPS intra-class scheduling algorithm. For this experiment, rtPS traffic is generated at an average rate of 800 Kbps. The experiment is performed with increasing number of SSs. The average throughput achieved as function of number of SSs is shown in figure 28. The result suggests that the proposed rtPS scheduler remains quite stable with increasing number of SS and hence it is scalable.

Video Streaming Using rtPS Class The simulation is performed to assess the performance of the proposed rtPS scheduler in scheduling realtime video traffic. We implemented a realtime video transmission mechanism in Qualnet that can send videos encoded in HEVC as discussed in section 5. To perform the experiment we used *videochat* video sequence obtained from the video trace library of Arizona State University [50].

The minimum reserved traffic rate is 20 Kbps and the maximum sustained traffic rate is 50 Kbps for each video stream. The experiment is performed by gradually increasing the number of videos transmitted simultaneously. Peak Service to Noise Ratio (PSNR) is used to compare the quality of the received video to the transmitted video. Figure 29 shows the average PSNR as function of number of concurrent video streams. The average PSNR is simply the arithmetic mean of PSNR of all video streams. High values of PSNR implies that the quality of received videos is good. In fact, the received videos have negligible distortions and are almost identical to the transmitted videos. In case of limited bandwidth, there is a possibility of transmitting a low-resolution image and reconstructing a high-resolution image from the transmitted low-resolution image at the receiver [53].

7 Conclusion

In this paper we have presented a two-level scheduling algorithm for the base station uplink scheduler for IEEE 802.16 standard. In the first level, bandwidth is allocated to different classes of traffic according to bandwidth demands and QoS requirements in terms of throughput, delay and fairness. Then in the second level, class-specific algorithms are used to distribute bandwidth among service flows of the same class. We have also developed a realtime video transmission mechanism to assess the performance of rtPS scheduler. The simulation studies show that the proposed solution is scalable and it ensures QoS for all classes of traffic supported by the standard, avoid starvation of lower priority flows and ensure fair bandwidth distribution. Furthermore, the video transmission mechanism performed efficiently and the quality of the received videos was good.

Acknowledgment

We would like to thank to Jeanpierre Guedon, Professor at the University de Nantes, for his valuable guidance and ideas on improving the algorithms proposed in this paper.

References

1. WG802.16 - Broadband Wireless Access Working Group, "IEEE Std 802.16.1-2012 - IEEE Standard for WirelessMAN-Advanced Air Interface for Broadband Wireless Access Systems," May 2012.
2. WiMAX Forum, "URL: http://www.wimaxforum.org," accessed on 18 Jan 2017.
3. B. B. Gupta, S. Yamaguchi, and D. P. Agrawal, "Advances in security and privacy of multimedia big data in mobile and cloud computing," *Multimedia Tools and Applications*, vol. 77, no. 7, pp. 9203–9208, 2018.
4. M. Ibtihal, N. Hassan *et al.*, "Homomorphic encryption as a service for outsourced images in mobile cloud computing environment," *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 7, no. 2, pp. 27–40, 2017.
5. S. Atawneh, A. Almomani, H. Al Bazar, P. Sumari, and B. Gupta, "Secure and imperceptible digital image steganographic algorithm based on diamond encoding in dwt domain," *Multimedia tools and applications*, vol. 76, no. 18, pp. 18 451–18 472, 2017.
6. B. Gupta, D. P. Agrawal, and S. Yamaguchi, *Handbook of research on modern cryptographic solutions for computer and cyber security*. IGI Global, 2016.
7. Y. Jararweh, M. Al-Ayyoub, M. Fakirah, L. Alawneh, and B. B. Gupta, "Improving the performance of the needleman-wunsch algorithm using parallelization and vectorization techniques," *Multimedia Tools and Applications*, pp. 1–17, 2017.
8. M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. Silva, F. Tufvesson, A. Benjebbour, and G. Wunder, "5g: A tutorial overview of standards, trials, challenges, deployment, and practice," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1201–1221, 2017.
9. Z. Ahmed and S. Hamma, "Efficient and fair scheduling of rtPS traffic in IEEE 802.16 point-to-multipoint networks," in *4th Joint IFIP/IEEE wireless and mobile networking conference*, Oct 2011.
10. Z. Ahmed and S. Hamma, "Two-Level Scheduling Algorithm for Different Classes of Traffic in WiMAX Networks," in *International Symposium on Performance Evaluation of Computer and Telecommunications Systems (SPECTS'12)*, July 2012.
11. IEEE 802.16e, "IEEE 802.16-2005, IEEE standard for local and metropolitan area networks - Part 16: Air interface for fixed and mobile broadband wireless access systems amendment for physical and medium access control layers for combined fixed and mobile operation in licensed bands," February 2006.
12. IEEE 802.16-2012, "802.16-2012 - IEEE Standard for Air Interface for Broadband Wireless Access Systems," August 2012.
13. L. Chaari, A. Saddoud, R. Maaloul, and L. Kamoun, "A comprehensive survey on wimax scheduling approaches," in *Quality of Service and Resource Allocation in WiMAX*. InTech, 2012.
14. E. Hahne and R. Gallager, "Round robin scheduling for fair flow control in data communication networks," in *IEEE International Conference on Communications*. IEEE, June 1986.
15. M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, "Weighted round-robin cell multiplexing in a general-purpose atm switch chip," *Selected Areas in Communications, IEEE Journal on*, vol. 9, no. 8, pp. 1265 –1279, oct 1991.
16. P. Goyal, H. M. Vin, and H. Chen, "Start-time fair queueing: a scheduling algorithm for integrated services packet switching networks," *SIGCOMM Comput. Commun. Rev.*, vol. 26, no. 4, pp. 157–168, Aug. 1996. [Online]. Available: http://doi.acm.org/10.1145/248157.248171
17. Y. Wang, S. Chan, M. Zukerman, and R. Harris, "Priority-based fair scheduling for multimedia wimax uplink traffic," in *Communications, 2008. ICC '08. IEEE International Conference on*, may 2008, pp. 301 –305.
18. W. Lilei and X. Huimin, "A new management strategy of service flow in ieee 802.16 systems," in *Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on*, june 2008, pp. 1716 –1719.
19. K. Wongthavarawat and A. Ganz, "Packet scheduling for QoS support in IEEE 802.16 broadband wireless access systems," *International Journal of Communication Systems*, no. 16, pp. 81–96, 2003.
20. J. Chen, W. Jiao, and H. Wang, "A service flow management strategy for IEEE 802.16 broadband wireless access systems in TDD mode," in *IEEE International Conference on Communications*, vol. 5, may 2005, pp. 3422 – 3426 Vol. 5.
21. C.-Y. Chang, M.-H. Li, W.-C. Huang, and S.-C. Lee, "An optimal scheduling algorithm for maximizing throughput in wimax networks," *IEEE Systems Journal*, vol. 9, no. 2, pp. 542–555, 2015.
22. M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round-robin," *Networking, IEEE/ACM Transactions on*, vol. 4, no. 3, pp. 375 –385, jun 1996.
23. H. Safa, H. Artail, R. Soudah, and S. Khayat, "New scheduling architecture for IEEE 802.16 wireless metropolitan area networks," *IEEE/ACM Transactions on Computer Systems and Applications*, pp. 203–210, 2007.
24. J. Chen, W. Jiao, and Q. Guo, "An integrated QoS control architecture for IEEE 802.16 broadband wireless access systems," in *IEEE Global Telecommunication Conference*, IEEE Communications Society, Ed., 2005.
25. X. Zhang, G. Zhang, and H. Sun, "A bandwidth allocation algorithm and its performance analysis based on IEEE 802.16d standard," in *7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'11)*, sept. 2011, pp. 1 –4.
26. Y. Shang and S. Cheng, "An enhanced packet scheduling algorithm for QoS support in IEEE 802.16 wireless networks," in *Third International Conference on Networking and Mobile Computing*, 2005, pp. 652–661.
27. J. Bennett and H. Zhang, "Wf2q: worst-case fair weighted fair queueing," in *INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, vol. 1, mar 1996, pp. 120 –128 vol.1.
28. C. Cicconetti, L. Lenzini, E. Mingozzi, and C. Eklund, "Quality of service support in IEEE 802.16 networks," *IEEE Network*, no. 20, pp. 50–55, 2006.
29. A. Sayenko, O. Alanen, and T. Hamalainen, "Scheduling solution for the IEEE 802.16 base station," *Computer Networks*, vol. 52, no. 1, pp. 96 – 115, 2008.

30. L. Chan, H. Chao, and Z. Chou, "Two-tier scheduling algorithm for uplink transmissions in IEEE 802.16 broadband wireless access systems," in *International Conference on Wireless Communications, Networking, and Mobile Computing*, 2006, pp. 1–4.
31. J. Ku, S. Kim, S. Kim, S. Shin, J. Kim, and C. Kang, "Adaptive delay threshold-based priority queueing scheme for packet scheduling in mobile broadband wireless access system," in *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, vol. 2, april 2006, pp. 1142 –1147.
32. T. Tsai and C. Wang, "Routing and admission control in IEEE 802.16 distributed mesh networks," in *IFIP International Conference on Wireless and Optical Communications Networks*, Singapore, 2007, pp. 1–5.
33. R. Fei, K. Yang, S. Ou, S. Zhong, and L. Gao, "A utility-based dynamic bandwidth allocation algorithm with QoS guarantee for IEEE 802.16j-enabled vehicular networks," in *Eighth International Conference on Embedded Computing SCALCOM-EMBEDDEDCOM'09*, sept. 2009, pp. 200 – 205.
34. S. Sengupta, M. Chatterjee, S. Ganguly, and R. Izmailov, "Exploiting MAC flexibility in WiMAX for media streaming," in *Sixth [IEEE] International Symposium World of Mobile and Multimedia Networks*, 2005, pp. 1–5.
35. U. K. Dutta, M. A. Razzaque, M. A. Al-Wadud, M. S. Islam, M. S. Hossain, and B. Gupta, "Self-adaptive scheduling of base transceiver stations in green 5g networks," *IEEE Access*, vol. 6, pp. 7958–7969, 2018.
36. A. Belghith and L. Nuaymi, "Comparison of wimax scheduling algorithms and proposals for the rtps qos class," in *Wireless Conference, 2008. EW 2008. 14th European*, ENST Bretagne, Rennes. Prague: IEEE, June 2008, pp. 1–6.
37. C. Ball, F. Treml, X. Gaube, and A. Klein, "Performance analysis of temporary removal scheduling applied to mobile wimax scenarios in tight frequency reuse," in *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*, vol. 2, sept. 2005, pp. 888 –894 Vol. 2.
38. P. Rukmani and R. Ganesan, "Adaptive modified low latency queuing algorithm for real time traffic in wimax networks," *Journal of Engineering Science and Technology*, vol. 12, no. 9, pp. 2551–2566, 2017.
39. R. Cruz, "A calculus for network delay. i. network elements in isolation," *Information Theory, IEEE Transactions on*, vol. 37, no. 1, pp. 114 –131, jan 1991.
40. V. Rangel, J. Ortiz, and J. Gomez, "Performance analysis of QoS scheduling in broadband IEEE 802.16 based networks," in *OPNETWORK Technology Conference*, Washington D.C., 2006.
41. D.-N. Lai, T.-C. Huang, and H.-Y. Chi, "Efficient bandwidth allocation with QoS guarantee for IEEE 802.16 systems," in *International Conference on Parallel Processing (ICPP)*, sept. 2011, pp. 115 –119.
42. A. Sayenko, O. Alanen, J. Karhula, and T. Hämäläinen, "Ensuring the QoS requirements in 802.16 scheduling," in *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems*. New York, NY, USA: ACM, 2006, pp. 108–117. [Online]. Available: http://doi.acm.org/10.1145/1164717.1164737
43. G. J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand *et al.*, "Overview of the high efficiency video coding(hevc) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.
44. "High efficiency video coding (hevc) test model 16 (hm 16) improved encoder description update 9."
45. H.265 : High efficiency video coding. [Online]. Available: https://www.itu.int/rec/T-REC-H.265
46. J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards"including high efficiency video coding (hevc)," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1669–1684, 2012.
47. P. Hanhart, M. Rerabek, F. De Simone, and T. Ebrahimi, "Subjective quality evaluation of the upcoming hevc video compression standard," in *Applications of Digital Image Processing XXXV*, vol. 8499. International Society for Optics and Photonics, 2012, p. 84990V.
48. K. Sharman and K. Suehring, "Common test conditions (jctvc-z1100)," *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11*, 2017.
49. E. N. Linzer and H.-M. Leung, "Context based adaptive binary arithmetic codec architecture for high quality video compression and decompression," Aug. 9 2005, uS Patent 6,927,710.
50. http://trace.eas.asu.edu/yuv/, accessed on 14th mar 2017.
51. Qualnet simulator (version 5.02). [Online]. Available: http://www.scalable-networks.com/products/qualnet/
52. "Qualnet 5.1: Advanced wireless model library," Scalable Network Technologies, Inc., September 2017.
53. H. Liu, Q. Guo, G. Wang, B. Gupta, and C. Zhang, "Medical image resolution enhancement for healthcare using nonlocal self-similarity and low-rank prior," *Multimedia Tools and Applications*, pp. 1–18, 2017.