Low Complexity Fair Scheduling in LTE Uplink Involving Different Traffic Classes

Atri Mukhopadhyay, Goutam Das

Abstract—Long term evolution (LTE) has already been accepted as the de-facto 4G wireless technology. However, the bulk of the research on LTE packet scheduling is concentrated in the downlink and the uplink is comparatively less explored. In uplink, mostly channel aware scheduling with throughput maximization has been studied. Further, channel aware scheduling requires an infinitely backlogged buffer model. This makes the investigations unrealistic. Therefore, in this work, we propose an LTE uplink packet scheduling procedure with a realistic traffic source. Firstly, we advocate a joint channel and buffer aware algorithm, which seeks to maximize the actual number of bits transmitted once a user is scheduled. Thereafter, we modify our algorithm to support traffic types with delay constraints. Next, we enhance our algorithm to support multiple classes of traffic. Finally, we have introduced priority flipping to minimize bandwidth starvation of lower priority traffics in presence of higher percentage of high priority traffic. In all the proposals, we have replaced the delay constraint by minimizing the packet drop due to delay violation. This further helps in reducing the problems to a well-known assignment problem.

Index Terms—Assignment, fairness, knapsack, LTE, MAC throughput, priority flipping, uplink scheduling.

1 INTRODUCTION

MOBILE communication has progressed in leaps and bounds over the past few years. The demand for high data rate from the mobile users has further fueled the advent of mobile networks. Long term evolution (LTE) by the third generation partnership project (3GPP) is one of the latest mobile technologies that provide data rates of the order to megabits per second [1]. LTE can provide seamless integration of voice, video and data. As a result, LTE and LTE-Advanced have become the most promising wireless access technologies.

LTE is fundamentally different from its preceding mobile wireless access technologies. LTE uses orthogonal frequency division multiple access (OFDMA) in its downlink and single carrier frequency division multiple access (SC-FDMA) in its uplink. OFDMA and SC-FDMA improve data rate by providing better interference management [1]. Both OFDMA and SC-FDMA divide the channel into multiple sub-carriers. However, scheduling the sub-carriers to different users such that the system utilization can be maximized is not a trivial task. Till now, downlink scheduling has been under the radar of most of the researchers while uplink scheduling is a comparatively less explored field.

In LTE, packet scheduling is the responsibility of the evolved nodeB (eNodeB). Packet scheduling refers to the act of allocating a certain group of sub-carriers for transmitting packets. Sub-carrier allocation is carried out in groups of 12 sub-carriers of 15 kHz each for a duration of 1 ms. This unit of allocation is known as the physical resource block (PRB). The time duration of 1 ms defines a transit time interval (TTI). Now, it is expected that each user experiences frequency selective fading in different PRBs. Further, different users experience different channel conditions on a certain PRB due to their different spatial positions. The channel conditions influence the modulation and coding scheme (MCS) that can be used on a certain PRB without violating the bit error constraints. Hence, the eNodeB has to allocate multiple PRBs to multiple users with the target of maximizing the possibility of bit transmission in each and every TTI. As already mentioned, scheduling is done on both the downlink and the uplink. LTE Downlink allows distributed allocation of PRBs [1][2]. However, in the LTE uplink, contiguous allocation of PRBs is more advisable due to the restrictions imposed by SC-FDMA. The contiguous PRB allocation restriction makes the assignment problem NP-Hard [3].

Further, when real-time packets with quality of service (QoS) constraints are scheduled, the scheduler has to worry about delay deadlines and packet drop constraints. In order to get around the problem of meeting delay deadlines, proposals with a two-stage scheduling are available in the literature [4][5]. The first stage is called time-domain packet scheduling (TDPS), while the second stage of scheduling is known as the frequency domain scheduling (FDPS). The TDPS shortlists the users based on head of the line (HoL) packet delay. Thereafter, the FDPS seeks to allocate the PRBs to the shortlisted users [4] with the target of maximizing the system throughput in terms of bits transmitted. Unfortunately, considering a subset of users for the final allocation results in suboptimal solutions [4]. In order to compute an optimal solution, one should investigate all possible options while keeping the computational complexity under control.

Moreover, most of the algorithms (for both downlink and uplink) existing in the literature consider infinitely backlogged model for data traffic and carry out only channel aware scheduling [3][4][6][7][8][9]. These models consider that packets are always available for transmis-
sion in the UE buffer. These algorithms, in fact, allocate the PRBs to the UEs that possess good channel gains even though the UEs may not have sufficient data in their buffer to utilize the allocated PRBs. However, the MAC throughput, which is the actual number of bits transmitted, may remain low as actual buffer status is not considered while scheduling the UEs [10]. Similar approach can be found in [11] where the authors have considered a buffer-based channel dependent scheduler (BCS) for two-hop relay-assisted LTE network. However, BCS is not an optimal algorithm and it does not guarantee contiguous channel allocation, which is a requirement as per LTE standards. In addition, UE has to send huge control information because the UE requires sending data in different non-contiguous PRBs with different modulation schemes.

In LTE scheduling, the final frontier is reached when we try to deal with traffic of multiple classes. Different classes of real-time traffic have different delay requirements. Moreover, data traffic, which is of best-effort type, has no delay requirements. The authors of [12] proposed two algorithms that emphasizes on QoS based resource allocations in LTE uplink. The proposals facilitate the packets of the ongoing flows to remain within delay deadlines while meeting a minimum rate constraint. However, the algorithms follow a greedy approach and hence do not provide optimum allocations. Two classes of traffic with different QoS requirements have been dealt with in [13] by switching between two heuristic algorithms. The work in [14] proposes a delay-aware algorithm that follows heuristic riding the peaks method of [3]. In [15], a standard compliant scheduling scheme for LTE-Advanced is discussed. The proposal describes a three stage adaptive and potential aware scheduling scheme (APASS) that carries out the PRB allocation for the traffic in a buffer while maintaining QoS. The final stage of APASS improves the scheduling performance of the initial allocation by following a potential zone concept. However, the user elimination method described in the second stage of APASS is heuristic and therefore, may result in suboptimal allocation. Further, APASS has a worst case complexity of \(O(U_{\text{max}}(U,N)^2 \log_2 N + NV)\), where \(N\) is the number of resource blocks, \(U\) is the number of active users and \(V\) is the number of schedulable users.

In this paper, we have developed an optimization problem for LTE uplink scheduling, which overcomes all the above difficulties with mathematical subtleties.

The paper is organized as follows. Section 2 provides motivation to the problem. Section 3 provides the details of the proposed delay aware packet scheduling algorithm. Section 4 sheds light on the delay and fairness aware traffic scheduling paradigm for mixed traffic. Section 5 provides a discussion on the algorithmic complexities of the proposals. Section 6 discusses the simulation model. Section 7 showcases the results and discussion followed by the conclusion.

## 2 Motivation

In this section, we discuss the issues related to channel aware scheduling that provide us the motivation to include buffer status within the same framework. Thereafter, we introduce real-time traffic streams scheduling and demonstrate that even channel plus buffer aware scheduling is not enough when traffic with delay deadlines are considered. We provide an example with three UEs that are competing for a single resource chunk (RC) to highlight the motivation for our proposed method. An RC is comprised of a set of six contiguous PRBs. In this example, we assume that the channel quality indicator (CQI) value does not change through the length of our example of five TTIs. In this example, we also assume that the number of bytes arriving per TTI to each of the users is deterministic and has a fixed value of 50. These assumptions are for the sake of maintaining simplicity. Later on, we prove through simulation that the conclusions drawn over here are rather generic. Table 1 provides the relationship between a certain CQI index value and the number of bytes of data that can be transmitted with the supported type of modulation. Table 2, on the other hand, illustrates the initial conditions.

As mentioned before, only channel aware scheduling, while neglecting the buffer conditions is not optimal and it induces unfairness. The UEs closer to the eNodeB are scheduled more frequently as they generally possess better channel gains. However, the UEs with poorer channel gains do not get enough transmission opportunities, which results in higher delay and packet loss [10].

### Table 1 Mapping of CQI to data transmission within a TTI

| CQI Index | Modulation | Data (1 PRB) | Data (1 RC=6PRBs) |
|-----------|------------|--------------|-------------------|
| 1-6       | QPSK       | 42           | 252               |
| 7-9       | 16-QAM     | 84           | 504               |
| 10-15     | 64-QAM     | 126          | 756               |

### Table 2 Example CQI and Buffer

| UE | CQI | Data transmission (possible bytes) | Arrival rate (bytes/second) | Initial buffer size (bytes) |
|----|-----|-----------------------------------|----------------------------|----------------------------|
| 1  | 7   | 504                               | 50                         | 400                        |
| 2  | 12  | 756                               | 50                         | 300                        |
| 3  | 6   | 252                               | 50                         | 260                        |

On the other hand, if buffer aware scheduling is used (see Table 3), the scheduling parameter becomes the minimum of the number of bytes that the channel condition allows to transmit for a particular UE and the number of bytes present in that UE’s buffer. This clearly increases fairness as the buffer occupancy also becomes a deciding factor. As a result, different users are selected in different TTIs (marked in bold). Hence, the overall system MAC throughput improves.

We present a few examples in the following tables to support our argument. The bold fonts in Table 3-5 indicate the scheduled users. The values in TTI(i+1) column of Table 3-5 indicate the updated values from TTI(i) column after accounting for the arriving and transmitted bytes. For example in Table 3, TTI(2) column value for UE1 is calculated by subtracting the number of bytes...
transmitted during TTI(1) (400) from the buffer content at the beginning of TTI(1) (400) followed by an addition of the number of newly arriving bytes (50).

In Table 4 and Table 5, we deal with traffic having delay deadlines. The number on the left side of the "/" indicates the buffer content and the number on the right side presents the number of bytes to be dropped if they are not transmitted in the current TTI. In this example, we assume that 20 bytes cross delay deadline in every TTI. Channel plus buffer aware scheduling is not very effective in this regard as can be seen in Table 4. The real-time packets have delay limits after which the packets are dropped. This may lead to a situation where the disadvantaged user suffers from heavy packet drop due to delay violation. Thus, the buffer content of these needy users may remain low and the scheduler may not allocate them bandwidth for data transmission. As a result, the system throughput over long term degrades and packet loss increases.

| Table 3: Channel plus Buffer Aware Scheduling |
|-----------------------------------------------|
| Buffer (bytes)                                | UE | TTI(1) | TTI(2) | TTI(3) | TTI(4) | TTI(5) |
| 1                                            |    | 400/50 | 50/0   | 100/50 | 150/50 | 200/50 |
| 2                                            |    | 300/50 | 350/0  | 50/0   | 100/50 | 150/50 |
| 3                                            |    | 260/250| 310/20 | 360/20 | 158/50 | 50/0   |

Therefore, to overcome this, packet delay can be a parameter to be further considered in such a scenario. However, minimizing delay (measured in seconds) and maximizing transmitted data (measured in bytes) makes the problem multi-dimensional. Further, putting a hard constraint on delay may lead to infeasibility when the number of delay constraint violating users is larger than the number of available resource units. This may result in non-functional scheduler. Hence, one needs to choose a parameter carefully to take care of both the problems mentioned above. We observe that the number of packets dropped if a user is not scheduled provides an elegant solution. Thus, we propose to maximize system throughput while simultaneously minimizing the overall packet drop. Since both the parameters are measured in bytes, a single objective function can be framed by subtracting the number of bytes that will be transmitted. This algorithm tries to schedule the most advantaged user suffers from heavy packet drop due to delay violation. Thus, the buffer content of these needy users may remain low and the scheduler may not allocate them bandwidth for data transmission. As a result, the system throughput over long term degrades and packet loss increases.

| Table 4: Channel plus Buffer Aware Scheduling for real time traffic |
|---------------------------------------------------------------------|
| Buffer/Packets to be dropped (bytes)                                | UE  | TTI(1) | TTI(2) | TTI(3) | TTI(4) | TTI(5) |
| 1                                                                  | 12  | 400/50 | 50/0   | 100/50 | 150/50 | 200/50 |
| 2                                                                  | 50  | 300/50 | 350/0  | 50/0   | 100/50 | 150/50 |
| 3                                                                  | 150 | 260/250| 310/20 | 360/20 | 158/50 | 50/0   |
| Total Transmitted                                                   |     | 400    | 250    | 100    | 150    | 420    |
| Total Drop                                                         |     | 355    | 5      | 20     | 20     | 420    |

| Table 5: Channel plus Buffer Aware Scheduling for real time traffic considering packet drop |
|-----------------------------------------------------------------------------------------------|
| Buffer/Packets to be dropped (bytes)                                                          | UE  | TTI(1) | TTI(2) | TTI(3) | TTI(4) | TTI(5) |
| 1                                                                  | 40  | 400/50 | 400/20 | 50/0   | 100/50 | 130/20 |
| 2                                                                  | 100 | 300/50 | 300/20 | 50/0   | 100/20 | 130/20 |
| 3                                                                  | 130 | 260/250| 260/25 | 55/5   | 100/20 | 130/20 |
| Total Transmitted                                                   |     | 252    | 400    | 280    | 130    | 192    |
| Total Drop                                                         |     | 153    | 25     | 20     | 20     | 238    |

Finally, to deal with multiple classes of real time traffic, the UE must judiciously use the allocated bandwidth. The existing schedulers in the UE employ strict priority scheduling where the voice queue is emptied first, followed by the video queue and the data queue respectively. This can lead to bandwidth starvation for the lower priority traffics, especially if the percentage of higher priority traffic is high in the traffic composition. Bandwidth starvation hinders the overall service quality. Hence, if the transmission of the higher priority traffic class can be safely delayed without violating its delay constraints in a certain TTI, the traffics of lower priority classes can be selected for transmission. We call this procedure as priority flipping. Priority flipping can significantly improve the user experience.

The contributions of this paper can be listed as follows:

1. We propose a delay aware real time scheduling algorithm that considers delay constraints for a single class of traffic while maximizing the number of bits transmitted. This algorithm tries to schedule the most deprived user.
2. Thereafter, we upgrade the algorithm for handling voice, video and data traffics that have different QoS requirements.
3. Finally, priority flipping has been introduced in the UE side to further enhance the QoS of lower priority classes while maintaining minimum QoS requirements of the higher priority class.

3 Delay Aware Real-Time Traffic Scheduling

In this section, we explain our previously proposed Dynamic Hungarian Algorithm with modification (DHAM) [10]. Thereafter, we extend DHAM to consider delay aware real-time traffic and to support QoS when only a single type of traffic is being transmitted. We list the symbols used in the following sections in Table 6.

| Symbols | Description                       |
|---------|-----------------------------------|
| N_u     | Number of active UEs.             |
| M       | Number of RCs.                    |
| Ω       | Set of delay violating users.     |
| N_r     | [10].                              |

Table 6 Symbols

3.1 MAC Throughput Maximization:

CA based scheduling is an elegant solution to scheduling problems in LTE upstream. However, one must remember that CA scheduling assumes infinitely backlogged model. Unfortunately, this assumption does not always hold in real world networks. Hence, along with CA, a buffer aware scheduling mechanism has to be introduced to enhance scheduling efficiency. We introduced such an algorithm (DHAM) in our previous work [10].

As explained in Section 2, for DHAM, one needs to prepare the channel gain matrix (C) and buffer status (B) as shown in Table 2.

Thereafter, P matrix is prepared, which signifies the number of bits that can be transmitted (refer Table 2) by mapping the elements of matrix C to values found by following the MCS given in Table 1. In our example, we have ignored the effect of coding and the corresponding P matrix values are represented in bytes.

Next, the elements of traffic matrix W is prepared as follows:

\[ w_{i,j} = \min(p_{i,j}, b_i) \]  

where, \( w_{i,j} \) is the element of traffic matrix W that corresponds to the \( i^{th} \) UE and the \( j^{th} \) RC; \( p_{i,j} \) is the element of matrix P that corresponds to the \( i^{th} \) UE and the \( j^{th} \) RC (maximum possible number of bytes that the \( i^{th} \) UE can send on the \( j^{th} \) RC) and \( b_i \) is the element of the vector B that corresponds to the \( i^{th} \) UE. In (1), the \( \min(p_{i,j}, b_i) \) is considered because even if \( p_{i,j} \) is greater than \( b_i \), the UE will not be able to send more data than \( b_i \) since, after sending \( b_i \), the UE will have no more data to send in the current TTI. On the contrary, if \( p_{i,j} \) is smaller than \( b_i \), the UE will be able to send at most \( p_{i,j} \) due to physical channel capacity constraints.

Thus, applying equation (1) over the example of Table 2, we get the final Traffic Matrix (W).

Utilizing the prepared W matrix, the scheduling optimization is performed with the help of Integer Linear Program (ILP) (2-5).

Maximize \[ \sum_{i,j} a_{i,j} w_{i,j} \]  
Subject to the constraints, \[ \sum_{i} a_{i,j} = M \] \[ \sum_{j} a_{i,j} \leq 1, \forall j \] \[ \sum_{i} a_{i,j} \leq 1, \forall i \]  

where, \( w_{i,j} \in W \) is obtained from (1); \( a_{i,j} \) is a binary variable, which is equal to 1 if UE \( i \) is allocated to RC \( j \). Otherwise, \( a_{i,j} = 0 \). The constraint (3) specifies that the number of allocations should be equal to the number of RCs present. The constraint (4) implies that an RC can be allocated to only a single UE. The constraint (5) signifies that one UE can get at most one RC.

In [10], we have shown that the scheduling problem falls under the category of the well-known assignment problem and Hungarian algorithm can be applied to solve it in polynomial time. However, it is to be noted that assignment problems require equal cardinalities of both the partites. Whereas, for the above mentioned scheduling, \( N_u \) is generally greater than \( M \). Therefore, one needs to use Hungarian algorithm after adding dummy RCs to transform the optimization problem to an assignment problem. The costs/rewards of connecting UEs to dummy RCs are set to zero. Therefore, the UEs that get allocated to dummy RCs are considered to be the UEs with no allocation in the current TTI.

3.2 MAC Throughput Maximization with QoS Support:

CA-Hungarian [7] and DHAM maximizes PHY throughput and MAC throughput respectively. DHAM works optimally when used with best-effort traffic. However, when we have traffic with delay constraints, i.e., real-time traffic, DHAM may inadvertently degrade the QoS. Examples of real-time traffic are voice and video. Since, DHAM is both CA and buffer aware, it is inclined towards scheduling the UEs with better channel quality. The users with poor channel quality may not get the opportunity for transmission even when their buffer builds up. Moreover, for real-time traffic, if a packet is not transmitted within its deadline, the packet is dropped. This results in further deterioration of the QoS. Hence, DHAM requires further modifications to ensure that it works efficiently for real-time traffic. The resulting modified protocol is termed as delay aware real-time scheduling (DARTS).

DARTS uses the same traffic matrix (W) used for DHAM. The criterion for scheduling a user using DARTS can be summarized using the following ILP.

Maximize \[ \sum_{i,j} a_{i,j} w_{i,j} \]  
Subject to the constraints, \[ \sum_{i} a_{i,j} \leq 1, \forall j \] \[ \sum_{j} a_{i,j} \leq 1, \forall i \] \[ \delta_i(1 - \sum_{j} a_{i,j}) \leq D_{th} - t, \forall i \]  

In the ILP (6-9), the constraints have inequalities in
order to accept unequal number of RCs and UEs. The constraint (7) means that an UE, if scheduled, can have at most one RC. Constraint (8) implies that an RC can be allocated to at most one UE. The constraint (9) is used to ensure that a real-time packet is scheduled within its delay constraints. Here, \( \delta_i \) indicates the delay of the head of the line (HoL) packet of the \( i \)th UE. If the UE has been scheduled in the current TTI then \( 1 - \sum a_{ij} = 0 \), otherwise it is equal to 1. \( D_{\delta_i} \) represents the delay deadline for the considered traffic and \( t \) is the length of the TTI. The entire inequality enforces scheduling of an UE when its buffer. Therefore, in order to enforce fairness, we need to minimize packet drop arising from the resulting scheduling. In another example that is shown in Table 8, we can clearly see that scheduling UE1 (more critical user) leads to a drop of \( 298+500=798 \) bytes. On the other hand, scheduling UE2 leads to a drop of 550 bytes. This will no doubt lead to slight unfairness, but it is better from the systems’ perspective. In other words, by scheduling UE2, we are avoiding future packet drop; thereby, controlling long term system throughput indirectly. DARTS ensures fairness over time and at the same time it enhances delay awareness without causing infeasibility.

Thus, in order to ensure fairness and minimize packet drop we need to modify the objective function of the optimization procedure used in DHAM. So, the new objective function becomes:

\[
\text{Maximize } \sum_i a_{ij}(w_{ij} - d_{ij}) - \sum_i \beta ik_i \quad (10)
\]

Subject to the constraints,

\[
\sum_i a_{ij} = 1, \forall j
\]

\[
\sum_i a_{ij} \leq 1, \forall i
\]

\[
\beta_i + \sum_j a_{ij} = 1, \forall i
\]

\[
a_{ij} = 0 \text{ or } 1
\]

\[
\beta_i = 0 \text{ or } 1
\]

where, \( k_i \) is the number of bytes that will be dropped if UE \( i \) is not scheduled in the current TTI and \( d_{ij} \) is the number of bytes that will be dropped by UE \( i \) if UE \( i \) is allocated to RC \( j \). Constraint (13) ensures that UE \( i \) is either allocated an RC \( \sum_j a_{ij} = 1 \) or the UE is mapped to the dummy RC \( \beta_i = 1 \). We replace \( w_{ij} - d_{ij} \) by \( \gamma_{ij} \) for compact representation. The problem described by Table 7 has been taken care of by the \( \sum_i \beta_i k_i \) term. On the other hand, consideration of \( \sum_i \sum_j a_{ij} \gamma_{ij} \) handles the problem described by the example in Table 8. The UE i can report the value of \( k_i \) along with the buffer status using a long buffer status report. It is impractical for the eNodeB to evaluate \( k_i \) as it will require the delay values of all the packets.

Hence, ILP (10-15) effectively replaces the delay constraint of ILP (6-9) by the packet drop metric. Now, looking closely at the objective function (10) and the constraint (13), we can identify that we can equate \( a_{i(M+1)} = \beta_i \) and treat \( y_{i(M+1)} = -k_i \). This modification makes the packets about to be dropped if not scheduled equivalent to a dummy RC and the dummy RC can have an inflow of \( (N_d - M) \). These \( (N_d - M) \) inflows are connected to the UEs that are not scheduled. For this discussion, we assume that \( (N_d > M) \). Interestingly, ILP (10-15) becomes equivalent to the famous transportation problem (ILP (16-20)) with the divergence of the last dummy RC node being equal to \( (N_d - M) \). So, the new formulation is given by ILP (16-20).

\[
\text{Maximize } \sum_i \sum_{j=1}^{M+1} a_{ij} \gamma_{ij} \quad (16)
\]

Subject to the constraints,

\[
\sum_i a_{ij} = 1, \forall j = 1, 2, \ldots, M
\]

\[
\sum_i a_{ij} = N_d - M, \forall j = M + 1
\]

\[
\sum_i a_{ij} = 1, \forall i
\]

\[
a_{ij} = 0 \text{ or } 1
\]

Constraint (18) implies that the dummy node can have an inflow of \( (N_d - M) \). The other constraints are same as ILP (2-5).

Finally, in order to convert ILP (16-20) to an assignment problem (ILP (21-24)), we have to replicate the dummy RC node to create \( (N_d - M) \) dummy RCs and restrict the inflow of each of the dummy RCs to 1. In graphical terms, we will create as many dummy nodes as required to make the cardinalities of both the parties equal. The cost of connecting to each of the dummy nodes will be same for a certain UE. Fig. 1 is a graphical repre-

### Table 7: Example Traffic: Case 1

| UE | Data transmission possible (bytes) | Bytes to be dropped |
|----|----------------------------------|---------------------|
| 1  | 252                              | 260                 |
| 2  | 504                              | 100                 |

### Table 8: Example Traffic: Case 2

| UE | Data transmission possible (bytes) | Bytes to be dropped |
|----|----------------------------------|---------------------|
| 1  | 252                              | 550                 |
| 2  | 504                              | 500                 |
sentation of the creation of the dummy RCs. We have shown the edges for only one UE for brevity. The reader should visualize that every node of the left partite is connected to every element of the right partite in similar manner. This formulation allows us to use the low complexity Hungarian algorithm to solve our problem.

\[
\begin{align*}
\text{Maximize} & \quad \sum_{i} \sum_{j} x_{ij} \gamma_{ij} \\
\text{Subject to the constraints,} & \quad \sum_{i} a_{ij} = 1, \forall j = 1,2,\ldots,M, (M + 1), \ldots,N_{U} \\
& \quad \sum_{j} a_{ij} = 1, \forall i \\
& \quad a_{ij} = 0 \text{ or } 1
\end{align*}
\]

where, constraint (22) captures the conditions of constraints (17) and (18) of ILP (16-20). We consider \( N_{U} > M \) where \( 1,2,\ldots,M \) are the actual RCs and \( (M + 1), \ldots,N_{U} \) denote the dummy RCs.

Finally, if \( N_{U} > M \), then all the \( \omega \in \Omega \) can not be scheduled in the current TTI. This will inevitably lead to packet drop for the users that are not scheduled. Therefore, in order to ensure that these users get higher priority in the subsequent TTIs, the number of bytes that are dropped at the end of the current TTI will be added to the obtained \( k_{i} \) for the ensuing TTI. For long term fairness enforcement, this packet drop history can be recorded for a window of the past \( n \) TTIs and the sum can be used in the \((n + 1)^{th}\) TTI. The procedure is given in (25)

\[
k_{i} = \sum_{\omega=1}^{\Omega} \epsilon_{i}
\]

where, \( \epsilon_{i} \) is the packet drop history for the \( i^{th} \) TTI.

Figure 1. Bipartites showing UE and RCs and the costs of connecting them. (a) without adding dummy RCs. (b) adding dummy RCs with same cost for each UE (assignment problem).

3.3 Applying DARTS when \( N_{U} < M \):

For the previous discussions in Section 3.2, we have considered that \( N_{U} > M \). However, another case may arise where \( N_{U} < M \). In such a scenario, if we allow only one RC per UE, some RCs will be wasted which could have been used by the active UEs. Therefore, we use ILP (10-15) in an iterative manner for this purpose. However, we omit the \( d_{ij} \) term for this purpose. The term \( d_{ij} \) is obtained through preprocessing by considering the fact that a certain UE will be allocated only a single RC. But, in this scenario, there is a scope for allocation of multiple RCs to a single UE. Hence, the calculation of \( d_{ij} \) becomes dependent on the allocations arising from multiple iterations. As a result, including \( d_{ij} \) will lead to sub-optimal allocations. Further, we prove through simulations that the effect of using \( d_{ij} \) is minimal in such a scenario.

Moreover, \( k_{i} \) is made equal to the number of bytes that will be dropped if UE \( i \) is not scheduled in the current TTI. We can clearly see that as long as the number of UEs is less than the number of RCs considered for allocation, the term \( \sum_{\omega} \beta_{\omega} k_{i} \) does not play any role in the allocation. At the end of every iteration, the values of \( k_{i} \) are updated by subtracting the number of bytes that will be transmitted by UE \( i \) due the allocation obtained in the current iteration. The minimum value that \( k_{i} \) can attain is equal to zero. In the final iteration, the updated value of the term \( \sum_{\omega} \beta_{\omega} k_{i} \) ensures that the delay violating UEs may be exclusively considered. It should be noted that since \( N_{U} < M \), dummy UEs with zero buffer content are added in this case for framing an assignment problem.

4 Delay and Fairness Aware Traffic Scheduling for Mixed Traffic

In this section, we propose a delay-aware fair scheduling algorithm for mixed traffic (DAFS) that maximizes the MAC throughput while meeting minimum QoS constraints when multiple traffic types are being scheduled. Thereafter, we propose priority flipping at the UE to further mitigate lower priority traffic starvation.

4.1 Delay-aware fair scheduling algorithm for mixed traffic executed at the eNodeB:

In Section 3, a single class traffic with delay deadline was considered. However, one must look into multiple classes of traffic because a single user may generate voice, video and data packets at the same time. These different types of traffic have varying QoS requirements. For example, voice generally has stringent delay bounds (less than 50 ms) for the air interface [5]. Video, on the other hand, has to satisfy relatively relaxed delay deadlines (less than 150 ms) [5]. Finally, data has no delay constraints. However, data packets may suffer from bandwidth starvation if the real time traffic load is sufficiently high. Hence, we need to take care of each of the traffic types individually in order to provide overall user satisfaction; which demands that we must feed parameters relating to each of the traffic types in the optimization engine. However, allowing too many parameters to enter into the optimization algorithm makes the convergence time impractical from the implementation point of view. Therefore, we construct a single parameter that ensures the individual delay bounds of each classes present in the considered traffic mix. This further keeps the execution time of the optimization algorithm within practical limits.

Our modified metric \( (k_{i}) \) for \( i^{th} \) user is given below,

\[
k_{i} = m_{i}^{v} + m_{i}^{d} + m_{i}^{q}
\]

where, \( m_{i}^{v} \) is the number of voice packets that will be dropped, if the user is not scheduled in the current TTI; \( m_{i}^{d} \) is the number of video packets that will be dropped if the user is not scheduled in the current TTI; and \( m_{i}^{q} \) is the number of bytes in the buffer that is above a pre-defined buffering threshold \( (B_{th} \text{ in bytes}) \).
4.2 DAFS with priority flipping at the UE:

In this sub-section, we mitigate lower priority traffic starvation. In contrast to the algorithms discussed so far, where the primary processing is done in the eNodeBs and UEs employ a strict priority packet forwarding, DAFS with priority flipping (DAFS-PF) employs additional optimization at the UE.

In strict priority transmissions, the order of priority followed is voice, video and then data. However, in strict priority transmission, if the higher priority traffic composition is sufficiently high then there is constant arrival of traffic in the higher priority buffers. This results in deferring of the lower priority packet transmission as the lower priority packets are transmitted only when all the higher priority traffic buffers are empty. Hence, the lower priority packets may suffer from starvation.

Therefore, in DAFS-PF, the UE may increase the priority value of the lower priority packets if it senses the development of a possible starvation. In order to achieve this, we associate rewards to packets and our target is to maximize the rewards while filling up the allocated bandwidth such that it is not over flowed. This requirement matches with the requirements of a knapsack problem. The problem is formally defined as follows.

\[
\text{Maximize } \sum_{i} \sum_{j} \alpha_{ij} r_{ij} \tag{27}
\]

Subject to the constraints,

\[
\sum_{i} \sum_{j} \alpha_{ij} w_{ij} \leq G \tag{28}
\]

\[
\alpha_{ij} = 0 \text{ or } 1 \tag{29}
\]

where, \( j \in \{ \text{voice, video, data} \} \), \( \alpha_{ij} \) is a binary variable indicating the scheduling information of the \( i^{th} \) packet of the \( j^{th} \) type, \( r_{ij} \) is the reward associated with the \( i^{th} \) packet of the \( j^{th} \) type, \( w_{ij} \) is the size of the \( i^{th} \) packet of the \( j^{th} \) type and \( G \) is the allocated bandwidth. Next, we explain the design of the reward variables that suits our purpose.

\[
r_{\text{voice}} = \frac{D_{\text{vvoic}}}{D_{\text{Thr,voic}}} \cdot r_{\text{video}} = \frac{D_{\text{vvoic}}}{D_{\text{Thr,v}}} \tag{30}
\]

\[
r_{i,\text{data}} = \begin{cases} 
\frac{B_{i} - B_{\text{Thr}}}{\text{Thr}_{i}} & \text{if } B_{i} > B_{\text{Thr}} \\
0 & \text{otherwise} \end{cases} \tag{31}
\]

where, \( D_{\text{vvoic}} \) is the \( i^{th} \) voice packet delay, \( D_{\text{vvoic}} \) is the \( i^{th} \) video packet delay, \( D_{\text{Thr,voic}} \) is the voice delay threshold, \( D_{\text{Thr,voic}} \) is the video delay threshold, \( B_{i} \) is the current buffer occupancy, \( B_{\text{Thr}} \) is the buffer threshold and \( B \) is the buffer capacity.

The explanations of the reward metrics are as follows: For both voice and video, as the delay of a packet approaches the maximum delay for which the packet can be stored in the buffer, the reward for transmitting the packet increases. Therefore, if video is getting starved due to high input of voice packets, at some point of time \( r_{\text{video}} \) will be greater than \( r_{\text{voice}} \) and hence the video packet will get transmitted prior to the voice packets.

However, data packets are not of the class of real time traffic and therefore, cannot be associated with delay. Hence, we choose buffer build up as the metric for prioritizing data packets. Since, data packets can be delayed, they are not given any priority when \( B_{i} < B_{\text{Thr}} \). However, when \( B_{i} > B_{\text{Thr}} \), and data packets are present in the buffer, we can say that the data packets are being starved due to the presence of other higher priority packets. Therefore, the data packets are marked and same amount of reward is associated to each and every marked data packet. The priority among the data packets follows first-in-first-out principle. Therefore, as the buffer gets filled up more and more, the priority of the data packets gets increased and they become more likely to be transmitted. Hence, DAFS-PF seeks to mitigate lower priority traffic starvation.

In order to minimize the execution time of the algorithm, these metric calculations must be a part of pre-processing before every scheduling instance. During the scheduling instance, only the knapsack algorithm is executed. The design of the algorithm will be such that all the metrics (rewards) will be recalculated while packet transmissions in the preceding TTI are ongoing. This helps in minimizing scheduling delay.

5 A NOTE ON COMPLEXITY

The creation of the traffic matrix involves \( O(mn) \), where \( m \) is the number of RCs and \( n \) is the number of UEs. For creating a symmetric matrix, the complexity becomes \( O(\max(m,n)^2) \).

The traffic creation matrix of DARTS and DAFS is \( O(m(n+1)) \). The assignment complexity is same as that of \( O(\max(m,n)^3) \). Thus, the complexity is \( O(\max(m,n)^3 + O(\max(m,n + 1)^2) \).

The user side action of DAFS-PF has to perform a user procedure following the knapsack optimization paradigm. The optimization solution has been obtained through the greedy method as there is packet fragmentation provision in LTE. The resulting complexity is \( O(n \log n) \).

6 SIMULATION MODEL

The simulation studies have been carried out on a system level simulator developed in the OMNeT++ network simulator. We have considered a seven cell scenario (Fig. 2), where a single cell is surrounded by six first tier cells. The studies have been carried out on the center cell, which acts as the serving cell and the first tier cells provide the interfering signal power. All the results obtained have been recorded in the center cell.

The UEs have been deployed in the cells by following a Poisson point process. In this paper, we have assumed that all the cells are directly connected to its serving eNodeB. Uplink transmission power control and MCS have been considered as per the guidelines provided in [16]. We have used a block fading channel model, where
the channel conditions remain constant over a TTI [11][19]. The work assumes that the UEs are not mobile. However, the work is perfectly valid for users with mobility. Further, we assume that the eNodeB has knowledge of the CQIs, buffer lengths and critical packets of all the UEs at the time of taking the scheduling decisions. For changing the input voice load, we have varied the generation interval of the packets of a two-state Markov VoIP Model [16]. Similarly, for the video model [17] we have altered the frames per second in order to vary the load. The load of the self-similar data source has been varied following the method given in [21]. The details of the simulation parameters are listed in Table 9 and Table 10 [16][19].

![Figure 2. Simulation Model](image)

| Parameter                        | Values |
|----------------------------------|--------|
| Traffic models                   |        |
| Voice traffic model              | two-state Markov [16] |
| Voice packet size                | 40 bytes |
| Silence indicator (SID) packet size | 15 bytes |
| Video traffic                    | Near real time video [17] |
| Frame per second                 | 15 |
| No. of packets in a frame        | 8 |
| Min. video frame size            | 1.5 kilo bytes |
| Packet size                      | Truncated Pareto; K=40 bytes, α=1.2, mean=50 bytes, max=250 bytes |
| Packet inter-arrival time        | Truncated Pareto; K=2.5 ms, α=1.2, mean= 6 ms, max=12.5 ms |
| Data traffic                     | Self-similar [18] |
| Packet payload                   | uniformly distributed between [46,1500] bytes |

### Table 10 LTE PARAMETERS

| Parameter                        | Value |
|----------------------------------|-------|
| Scenario                         | UMa   |
| Inter-site distance              | 500 m |
| System bandwidth                 | 10 MHz |
| Center frequency                 | 2 GHz |
| No. of prbs (N_{PRB})            | 50 (48 for data) |
| No. of prbs in a RC              | 6     |
| Path loss (PL) model             | Non line of sight |
| Shadowing standard deviation     | 4 dB |
| UE max transmit power (P_{max})  | 24 dBm |
| Uplink power control (PC)        | max(P_{max}, P_s + aPL10log_{10}N_{PRB}) |

### Results and Discussion

This section illustrates the performance of the proposed algorithms, namely DARTS, DAFS and DAFS-PF. In our previous work, we have extensively studied DHAM [10]. In that work, we have compared the performance of DHAM with that of recursive maximum expansion [6], channel aware Hungarian algorithm [7] and buffer based channel dependent scheduler [11]. In [10], we have established that DHAM provides better MAC throughput, buffer build up and fairness as compared to the other scheduling protocols. Therefore, in this paper, we compare our newly proposed algorithms like DARTS, DAFS and DAFS-PF with DHAM only.

#### 7.1 Delay aware real-time scheduling:

In this sub-section, we compare the performance of DARTS (introduced in Section 3.2) with DHAM. The main parameters to study are the fairness, the MAC throughput, the number of packets delivered by the user having worst channel conditions and the delay.

DARTS performs much better in terms of fairness as depicted in Fig. 3a. Fairness is improved in case of DARTS because whenever a user suffers from bad channel conditions for a considerable amount of time, DARTS provides opportunity for them to occupy the channel. Whereas, DHAM only seeks to improve the MAC throughput and as a result generally selects the users with better channel quality. However, it should be noted that the improvement on fairness is obtained without any sacrifice on MAC throughput as seen in Fig. 3b. This happens as DARTS reduces packet drop due to delay.

The scheduling principle of DARTS prioritizes users who have packets about to violate the delay constraints. Now, if the input rates of all the users are similar, the poor channel quality users are normally worst hit. Fig. 4a, illustrates the enhancement in the delivery of packets for a user having the worst channel condition due to the delay aware scheduling of DARTS.

![Figure 3. (a) MAC Fairness vs Network Load (a) MAC Throughput vs Network Load](image)
Finally, in order to prefer the users with higher number of delayed packets, DARTS defers the transmission from the users with better channel quality and therefore increases the average delay. However, DARTS ensures that the delay remains within the delay limits. Fig. 4b shows the effect of DARTS on mean voice delay.

7.2 QoS enhancement of multiple classes of traffic:

This sub-section displays the effects of the DAFS and DAFS-PF when multiple classes of traffic are present in the traffic mix. DAFS is the multiclass version of DARTS and therefore, we have not included the results of DARTS in this sub-section. DAFS and DAFS-PF has been compared against the performance of DHAM and APASS. Note that, for the result compilation, we have considered the DAFS version with packet drop history (sum of the voice and video metric for 1000 TTIs of recent past). For each of the following figures, we have three sub-plots [a, b and c]. For sub-plot (a), the voice load is made variable while the video and data load are fixed at 1Mbps. For the sub-plot (b), the video load is altered while the voice and data load are fixed at 1Mbps. Finally, for the last sub-plot (c), voice and video load are fixed to 1 Mbps, while data load is varied.

As seen in Fig. 5, the MAC fairness provided by DAFS and DAFS-PF is higher than that of both DHAM and APASS. This happens because, both DAFS and DAFS-PF give more priority to the users that have poorer channel conditions. APASS, on the other hand, removes schedulable users in a heuristic manner in its second stage. As a result, the optimal MAC fairness is not attained. However, enhancement of MAC fairness results in degradation of MAC throughput as can be seen from Fig. 6. MAC throughput is higher for DHAM as it seeks to maximize MAC throughput. However, DAFS and DAFS-
PF seeks to maximize MAC throughput while meeting delay constraints. As a result, the MAC throughput delivered by them is comparatively lower. However, the semi-heuristic nature of APASS makes it sub-optimal even in terms of MAC throughput.

Fig. 7 reveals that the DAFS leads to higher video packet drop as compared to DHAM when the percentage of voice traffic is higher in the traffic composition (see Fig. 7a). This directly follows from the fact that DAFS yields lower MAC throughput. Moreover, strict priority scheduling clears out the voice buffer before transmitting the video packets. As a result, video buffer gets filled up quickly when the voice traffic load is high. DAFS-PF tries to rectify this shortcoming of DAFS by the application of priority flipping. However, DAFS-PF also yields lower MAC throughput than DHAM and hence drops higher number of video packets as compared to DHAM. However, the performance of APASS is poorer than all of our proposals as it ignores several bandwidth starving users because of the usage of heuristic utility functions. On the other hand, when the percentage of voice is low in the traffic mix, DHAM, DAFS and DAFS-PF gives comparable performance (see Fig. 7b and 7c).

Figure 8. Number of worst user video packets delivered (a) Voice load variable, video and data fixed to 1 Mbps (b) Video load variable, voice and data fixed to 1 Mbps (c) Data load variable, voice and video fixed to 1 Mbps.

However, when it comes to the number of video packets transmitted by the worst channel user, DAFS and DAFS-PF clearly out performs DHAM. We observe that in this regard, DAFS-PF performs the best because of priority flipping. The comparison can be seen in Fig. 8. The number of video packets delivered by the worst user decreases as load increases. This is because lack of the transmission opportunities increases the number of packet dropped. This happens as all the UEs accumulate large number of packets in their queues when the load is high. However, the video performance of the DAFS-PF is improved by borrowing the bandwidth that was to be used for transmitting voice packets in DAFS. Hence, the worst user voice performance is poorer for DAFS-PF than it is for DAFS as shown in Fig. 9. Nevertheless, both the algorithms perform better than DHAM for heavy load conditions because of their fair scheduling policy. APASS, however, shows significantly poor scheduling for the worst channel user. The primary reason for this is because of the heuristic utility function used in APASS. We believe that APASS is inefficient in a static scenario as the mean path-loss of the users do not change in such a situation. However, our optimal choice of the utility function makes it efficient in a static scenario.

Finally, an interesting trend can be observed in Fig. 9b and Fig. 9c. It may be noted that the number of voice packets delivered when video or data load is varied while keeping the voice load fixed falls till the network load of 5Mbps (Fig. 9b and Fig. 9c). At this moderate load, the worst user does not get enough selection preference as the voice packets are dropped due to delay overshoot. However, as the load increases further, the fairness is invoked due to higher accumulation of video and data packets. Finally, as the load increases even further, all the users become heavily loaded and compete for the available bandwidth. Therefore, the voice packet reception from the worst user decreases again.

**CONCLUSION**

In this paper, we have demonstrated that channel aware scheduling alone in the LTE uplink should not be the de-facto mechanism for resource allocation. In order to utilize the physical resources efficiently, one must make a cross layer optimization including information from the MAC layer. The use of buffer state information
while scheduling is obligatory to increase the effective throughput, i.e., the MAC throughput.

However, this crude consideration may be rendered unfair when real-time traffic is used. Therefore, we have proposed DARTS for a single class of real time traffic, DAFS and DAFS-PF for multiple classes of real-time traffic. These advanced low complexity scheduling algorithms produce optimal results while successfully enhancing the fairness among users. The novelty of the algorithms lies in the usage of packet drop due to delay constraints in place of head of the line delay. Further, these algorithms make use of the long buffer status reports to transmit the packet drop value from the UEs to the eNodeB. This relieves the system from transmission of heavy control information. Finally, priority flipping, introduced in DAFS-PF, boosts video performance and reduces data starvation in appropriate scenarios.

Extensive simulation studies have confirmed that the LTE uplink performance can be improved significantly by incorporating the proposed schedulers in the LTE eNodeB.

REFERENCES

[1] F. Capozzi, G. Piro, L. A. Grieco, G. Boggia and P. Camarda, "Downlink Packet Scheduling in LTE Cellular Networks: Key Design Issues and a Survey," IEEE Communications Surveys & Tutorials, vol. 15, no. 2, pp. 678-700, Second Quarter 2013.
[2] N. Abu-Ali, A. E. M. Taha, M. Salah and H. Hassanen, "Uplink Scheduling in LTE and LTE-Advanced: Tutorial, Survey and Evaluation Framework," IEEE Communications Surveys & Tutorials, vol. 16, no. 3, pp. 1239-1265, Third Quarter 2014.
[3] S. B. Lee, I. Pefkianakis, A. Meyerson, S. Xu and S. Lu, "Proportional Fair Frequency-Domain Packet Scheduling for 3GPP LTE Uplink," Proc. IEEE INFOCOM 2009, Rio de Janeiro, 2009, pp. 2611-2615.
[4] A. Pokhariyal et al., "HARQ Aware Frequency Domain Packet Scheduler with Different Degrees of Fairness for the UTRAN Long Term Evolution," Proc. IEEE VTC-Spring, Dublin, 2007, pp. 2761-2765.
[5] B. Palit, SS Das, "Performance Evaluation of Mixed Traffic Schedulers in OFDMA Networks," Springer Wireless Personal Communications, vol. 83, no. 2, pp. 895-924, June 2015.
[6] L.A.M.R. de Temino, G. Berardinelli, S. Frattasi and P. Mogensen, "Channel aware scheduling algorithm for SC-FDMA in LTE uplink," Proc. of IEEE PIMRC 2008, pp.1-6, Sept. 2008.
[7] G.A. Medina-Acosta and J.A. Delgado-Penín, "On the feasibility of a channel-dependent scheduling for the SC-FDMA in 3GPP-LTE (mobile environment) based on a prioritized-bifacet Hungarian method," EURASIP Journal on Wireless Communications and Networking, 2011, vol. 2011:71.
[8] F.D. Calabrese, P.H. Michaelsen and C. Rosa, "Search-Tree based Uplink Channel Aware Packet Scheduling for UTRAN LTE," Proc. of IEEE VTC’08, pp.1949-1953, Singapore, May 2008.
[9] S. J. Lim, J. Chung and J. S. Jang, "Uplink scheduling algorithm for capacity enhancement in LTE," Proc. of 24th Telecommunications Forum (TELFOR), Belgrade 2016, pp. 1-4, Nov. 2016.
[10] A. Mukhopadhyay, G. Das and V. Sudheer Kumar Reddy, "A fair uplink scheduling algorithm to achieve higher MAC layer throughput in LTE," Proc. IEEE ICC, London, 2015, pp. 3119-3124.

[11] M. Mehta, S. Khakurel and A. Karandikar, "Buffer Based Channel dependent Uplink Scheduling in Relay-Assisted LTE Networks", Proc. of IEEE WCNC, pp.1777-1781, Shanghai, April 2012.
[12] O. Delgado and B. Jaumard, "Scheduling and resource allocation for multiclass services in LTE uplink system," Proc. 2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications, Niagara Falls, ON, 2010, pp. 355-360.
[13] F.-C. Kuo, K.-C. Ting, H.-C. Wang, C.-C. Tseng and M.-W. Chen, "Differentiating and Scheduling LTE Uplink Traffic Based on Exponentially Weighted Moving Average of Data Rate," Mobile Networks and Applications, vol. 22, no. 1, pp. 113-124, Feb. 2016.
[14] B. Amarasekara, C. Ranaweera, R. Evans and A. Nirmalathas, "Dynamic scheduling algorithm for LTE uplink with smart-metering traffic," Transactions on Emerging Telecommunications Technologies, Feb. 2017.
[15] A. Ragaleux; S. Baey; M. Karaca, "Standard-compliant LTE-A Uplink Scheduling Scheme with Quality of Service," IEEE Transactions on Vehicular Technology, vol.PP, no.99, pp.1-1, Jan. 2017.
[16] Report ITU-R M.2135-1 Guidelines for evaluation of radio interface technologies for IMT-advanced.
[17] 3GPP TR 25.892 Technical Report 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; "Feasibility Study for Orthogonal Frequency Division Multiplexing (OFDM) for UTRAN enhancement" (version 6.0.0 Release 6, 2004-06)
[18] W. Leland et al., "On the Self-Similar Nature of Ethernet Traffic (Extended Version)," IEEE/ACM Transactions on Networking, vol. 2, no. 1, Feb. 1994, pp. 1-15.
[19] 3GPP TS 36.213, “LTE: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures” (version 8.8.0 Release 8, 2009-10)
[20] A. Goldsmith, Wireless Communications. Cambridge Univ. Press, 2005.
[21] G. Kramer, B. Mukherjee, G. Pesavento, “IPACT: A Dynamic Protocol for Ethernet PON (EPON),” IEEE Communications Magazine, vol. 40, no. 2, pp. 74-80, Feb. 2002.

Atri Mukhopadhyay received his M.Tech degree in Distributed and Mobile Computing from Jadavpur University, India. He is presently working towards Ph.D. from GSSST, Indian Institute of Technology, Kharagpur, India. His research interests include wireless-optical integrated networks, optical access networks and wireless access networks.

Goutam Das has obtained his Ph.D. degree from the University of Melbourne, Australia in 2008. He has worked as a postdoctoral fellow at Ghent University, Belgium, from 2009-2011. Currently he is working as an Assistant Professor in the Indian Institute of Technology, Kharagpur. His research interest is in the area of optical access networks, optical data center networks, radio over fiber technology, optical packet switched networks and media access protocol design for application specific requirements.