Abstract — This report gives an overview of basic scheduling techniques and their application in Long Term Evolution (LTE). The second part deals with Hybrid-Automated Repeat Request (HARQ), the error correction algorithm used in LTE. The intriguing advantages of this error correction as well as its implementation in LTE are discussed.

1 Introduction

In every network that is accessed simultaneously by several users, it is necessary to coordinate the transmission to avoid collisions. An early approach to multi-user access was the experimental ALOHA protocol, where each user just accesses the channel whenever data should be sent. If a collision is detected, the data will be resent after a random time. This approach works; however, the capacity of the channel is very low. Therefore, better algorithms are needed to share the resources more efficiently. Some of the basic so-called scheduling algorithms are presented in chapter 2.

If a packet was not received successfully due to interference, noise or shadowing, error correction algorithms are mandatory to sustain an error-free transmission of data. In Long Term Evolution (LTE) this will be done by the Hybrid Automated Repeat Request (HARQ) protocol. The benefits of using HARQ and its mode of operation and implementation will be presented in the third chapter of this report.
2 Scheduling

As already stated in the introduction, the common channel has to be shared by multiple users and it is desirable to find an efficient solution to share the commonly used channel. A very simple example to avoid the problem of collisions is to assign each user its own channel. However, normally this is far from the optimal solution and usually it is not even possible to implement this due to technical limitations. Therefore we need more sophisticated approaches to divide shared resources between users.

These resources might be space in a Space Division Multiplexing system, code in Code Division Multiplexing systems, or time in Time Division Multiplexing systems and frequency in Frequency Division Multiplex systems. The Orthogonal Frequency Division Multiplexing (OFDM) modulation scheme used in the downlink of LTE is a possible approach to divide the bandwidth of a channel between different users. The point of scheduling is to find an acceptable solution how to divide the resources. Several different and partially contradictory aspects have to be considered by the scheduling algorithm. Some of them are:

- The maximal acceptable delay for a packet
- Fairness
- "User starvation" has to be avoided
- The algorithm has to allocate the resources efficiently
- The resulting overhead must be minimised

Three different scheduling algorithms will be examined and compared in several aspects. These three algorithms are

1. Round Robin
2. Maximum Rate
3. Proportional Fair

2.1 Round Robin Scheduler

The Round Robin Scheduler is the simplest known scheduling algorithm. It allocates equally spaced slots to each user. In a basic implementation, the Round Robin Scheduler allocates each user a time interval in which this single user gets exclusive access to the channel. As shown in Figure 1, the selection of the user who will get access is done in a Round Robin manner which means that each user will be placed in a queue. If everyone in the queue has been served, the scheduler will start again with the same queue. This scheduler will not consider any other information than the time for its scheduling decisions. Therefore, the implementation is very simple, but the achieved results are suboptimal.
The Round Robin algorithm is not efficient when considering the overall data rate in the cell. Only the selected user can transmit data and the data rate of this user is limited by the user’s actual Signal to Noise Ratio (Shannon law). If a user with a bad Signal to Noise Ratio (SNR) is selected, the data rate of the cell in this slot will be lower and therefore also the overall data throughput decreases. Normally, the SNR is varying over time and therefore the basic idea of other scheduling algorithms is to exploit this characteristic.

### 2.2 Maximum Rate Scheduler

The Maximum Rate Scheduler is the crudest approach to exploit the fluctuations of the SNR with the goal to obtain a higher overall data throughput in the cell. As shown in Figure 2, the Maximum Rate Scheduler simply selects the user whose SNR is highest and therefore the expected data rate is the highest of all users. If the SNR values of all users were equally distributed over time, every user would get the same transmission time. In this case, however, it would not be a random data rate as with the Round Robin Scheduler but always the highest present data rate in the cell. Therefore the overall data rate of the cell will be the theoretical maximum data rate.
This scheduler is very efficient, because it yields the highest cell data rate. However, the scheduler has severe drawbacks which make a practical application pointless.

Figure 3 shows what happens if, for example, a user is situated next to the base station and therefore has a much better signal quality than every other user in the cell. The scheduler will always schedule this single user for transfer and if its buffer is always full (for example during the download of a large file or video from the Internet), no other user will be scheduled for transmission and the connections of all other users in the cell will encounter a timeout.

To avoid massive user starvation as in this example, modifications to the scheduling algorithm are necessary. It is possible to define a minimal rate $r_{\text{min}}$ for each user and allocate this rate in a similar manner as the Round Robin Algorithm does. Only if every user has at least received $r_{\text{min}}$, other users will be scheduled according to the Maximum Rate Algorithm. With this modified algorithm, user starvation is quite unlikely. However, the overall data rate also decreases.

Another modification that might be applied to the practical implementation of this scheduler has the purpose to prevent it from changing users too often. Since the SNR can have small and fast changes, the scheduler might change very often between two users that have almost the same average SNR over a small interval of time. This will create a lot of overhead due to the constant switching between users. There are two ways to achieve this: only change the user after a defined interval, or use a hysteresis curve to determine the point when to pass the transfer token to another user.

2.3 Proportional Fair Scheduling

Another approach to scheduling is the Proportional Fair Scheduler. The basic idea of the Proportional Fair algorithm is to exploit the variations in channel quality as well, but this time the scope is not fixed on the absolute SNR but on the actual SNR of each user in relation to its average SNR value.

To run a Proportional Fair Scheduler, the eNodeB needs to measure the signal quality of each user’s channel independently and calculate the mean SNR. Usually, this is done by calculating
a moving mean of the SNR. The active user \( \hat{i} \) is chosen according to \( \hat{i} = \arg \max_i \left\{ \frac{r_{i,n+1}}{\theta_{i,n}} \right\} \). The mean SNR is represented by \( \theta_{i,n} \) where \( i \) is the identification number of each user and \( n \) is the time interval. \( r_{i,n+1} \) indicates the expected data rate for the next time interval for the user \( i \). Usually, the estimated data rate is derived from the actual SNR. A more sophisticated prediction of the expected data rate is normally not possible to obtain since the fast fluctuations in the link quality are unpredictable.

From these values, the eNodeB has to calculate the ratio for each user and then compare the ratios to determine the user with the highest ratio to schedule him for transfer in the next interval. The principle of a Proportional Fair Scheduler is illustrated in Figure 4. The overall throughput of the Proportional Fair Scheduler is usually higher than the output of a cell that is managed by a Round Robin Scheduler, because the users are not chosen in a fixed order. Instead, the user whose data rate is farthest above its average rate will be selected. Therefore, relatively to its transmission time, each user will be able to transmit more data than in a Round Robin scenario.

However, the throughput will be lower than the maximal throughput of a Maximum Rate Scheduler. Nevertheless, the Proportional Fair Scheduler introduces some degree of fairness and user starvation is less probable compared to the Maximum Rate Scheduler. The Proportional Fair Scheduler will also schedule users with very poor signal conditions for data transfer, but only when their SNR is at a peak. The obvious disadvantages of this scheduler are its high complexity, because for each user the ratio between predicted and average SNR has to be calculated and measured continuously. There are also scenarios that might yield an unwanted behaviour of this scheduler. When a user from a far distance moves closer to the base station in a scenario without signal shadowing (an area without buildings and flat terrain for example), its SNR will increase constantly and therefore the actual SNR will always be above its average. Hence, it is very probable that this user will be scheduled very often. On the other hand, if the user is moving away from the base station, its actual SNR will always be below its average SNR and therefore, it is improbable for it to be scheduled. This might lead to “starvation” of this user because its connections will timeout without any further transmissions.
2.4 Comparison

A good way to compare the behaviour of different scheduling algorithms with respect to fairness, is the cumulative distribution function (CDF) as shown in Figures 5 and 6. The CDF can be understood as the probability that every user in the cell will be served. This of course depends on the overall data rate in the cell which is referred to as “User Throughput” in these figures. It is obvious that with an overall data rate of zero it is impossible to serve any user, hence the probability that all users will be served is zero. The ascent of the Round Robin Scheduler is very high, since it equally distributes the time among the users and therefore is almost independent from the overall data rate.

The other schedulers show a more complex curve. It can be seen that it is very hard to serve each user with the Maximum Rate Scheduler. If, for example, a user is situated very far away from the eNodeB and his signal to noise ratio is very poor, it is very unlikely that it will be served at all and therefore a high overall data rate in the cell is needed for the scheduler to allocate a small slot for transmission to this user. This underlines the difficulties in the practical application of a Maximum Rate Scheduler without a minimal data rate for each user. The most obvious difference between the two scenarios can be observed at the Maximum Rate Scheduler. In a scenario with full buffers, where there are always data to transmit, the users with the best signal quality will be transmitting all the time and the other users will not be able to transfer data at all. In the Web browsing scenario, the

Figure 5: Several scheduler’s behaviour in a full buffer scenario [Kon09]

Figure 6: Several scheduler’s behaviour in a web browsing scenario [Kon09]
buffers are empty after a certain amount of data have been transmitted. Since these users will not transfer any additional data, the user with the next lower signal quality will be chosen. The behaviour of the Round Robin and the Proportional Fair Scheduler are almost unaffected by the traffic scenario because they already provide a certain degree of fairness and the probability that a users will completely be ignored is low compared to the Maximum Rate Scheduler [Kon09].

2.5 Remarks

For the implementation in LTE some special aspects have to be considered. The most important aspect is that the scheduling is done in the frequency domain. Different subcarriers are scheduled to different users due to the use of OFDM in the downlink and Single Carrier - Frequency Domain Multiple Access (SC-FDMA) in the uplink channel. The scheduling has to be done independently for each subcarrier, because the SNR might vary for different frequencies and the buffer-level has to be considered. If there are not enough data waiting for transfer, not all subcarriers should be allocated to one user. In the uplink channel, the transmission power of the UE is limited and therefore it would be inefficient to allocate all subcarriers to one UE [Dahl08].

Another important point of the scheduler implementation is the consideration of Quality of Service. Different users might get different priorities and different services might also be treated differently. Voice data for example is very sensitive to the delay of its transmission, but its data rate is relatively low. On the other hand, video streaming from the Internet for example is not critical to delay, but its data rate is high. Hence, it is desirable to give the voice data a higher priority than the video download.

It might also be desirable to treat several users differently. Some users might have different contracts with the service provider where they pay a higher fee to obtain a higher quality of service and therefore, their packets should be scheduled with higher priority.

In the following section of this paper, the error correction mechanism HARQ will be discussed, which causes previously flawed packets to be retransmitted. These retransmissions should be sent as fast as possible to avoid large delays of packets that were received erroneously by the receiver. Hence, these packets should be handled by the scheduler with high priority.

Another important point is that schedulers and scheduling algorithms are not part of the LTE specifications. In practise, schedulers are much more sophisticated (for example [Kwa08] gives better performance but is much more complex) than the algorithms presented in this paper and normally they are kept secret by the manufacturers [Dahl08]. Of course some interfaces have to be specified, for example how to obtain information about the channel quality of each user who is logged in to the cell.
3 HARQ

3.1 ARQ & FEC

As in every communication system, the transmission channel of LTE is also subject to errors. To limit these errors to an acceptable value, error correction mechanisms have to be applied to the transmission.

The errors can be divided into two basic groups, distinguished by their origin. The first group of errors is characterised by relatively slow variations of the channel quality and therefore a slowly varying bit error rate of the channel. These variations in channel quality might occur due to signal shadowing or an increasing distance between the base station and the mobile terminal. With a weakening signal, the SNR will increase and therefore the receiver will encounter more erroneous transmissions. This error pattern can be distinguished from errors that occur due to fast fluctuations in the signal quality and therefore in the bit error rate. These fluctuations are a result of intracell interference or receiver noise for example. These errors are unpredictable and normally only present for a short period of time. Because of the different error patterns, it is necessary to employ different countermeasures.

In LTE this is done by using two different ways to deal with errors in the transmission channel. The slow variations in channel quality will be compensated by Link Adaption, while the errors due to fast variations in channel quality will be corrected by the Hybrid-ARQ algorithm. HARQ stands for Hybrid Automated Repeat reQuest and as the name suggests, it is a hybrid form of two separate error correction schemes. The first approach to error correction is Forward Error Correction (FEC). Basically, FEC adds redundancy to the transmitted data by adding parity bits. Using these parity bits, a certain amount of erroneously received bits can be corrected at the receiver’s decoder. But of course, the transmission of parity bits is the price that has to be paid to gain the lower error probability. Since only a certain number of errors can be corrected with the number of transmitted parity bits, a high number of parity bits would be necessary to correct a high number of errors that might occur under bad circumstances.

The other correction mechanism is Automated Repeat reQuest, which means that the receiver has to perform an integrity check on the received packet. Normally this is done by a Cyclic Redundancy Check [Sol04]. If the received packet is error-free, it will be accepted and the next packet is expected. If an error occurs, the sender will be informed and performs a retransmission of the erroneously received datablock. For the ARQ error correction, only a small redundancy in the transmitted bits is needed, because it is only necessary to detect errors, not to correct any of them. The correction is done by retransmission of the packet, but the retransmission takes additional time and the block can only be delivered by the receiver’s decoder if all bits are error-free. Therefore, only a single flawed bit in the received block will lead to a retransmission of the whole block. This will dramatically increase the delay time of the channel and is therefore not desirable. This problem can be counteracted by shortening the length of the data blocks, but this might introduce other problems, for example an increased amount of overhead data and hence also a reduced payload data rate of the channel.

Therefore, a good way of error correction is to use both mechanisms at once to obtain the best
of both worlds. In HARQ, the FEC is used to correct a certain number of errors in the received packets. This amount of correctable errors depends on the desired code rate and has to be a trade-off between high data rate and low Bit Error Rate (BER) [Sol04]. However, the FEC does not have to be able to correct all errors under bad reception conditions because there is the ARQ layer which can handle packets with more errors than the FEC can correct. This leads to a higher data rate, because a larger part of the link capacity can be used for payload and does not have to be used for the parity data.

If a packet with a higher amount of errors than the FEC can handle arrives at the receiver, the ARQ will take over and correct the received errors. This leads to a higher delay for this corrupted packet, but only for this packet. Since this is not the expected case and should only happen for bad reception conditions, the increased delay for this single packet is preferred to a lower data rate for all packets if the redundancy would have been increased.

### 3.1.1 Chase Combining

The basic mode of operation of a simple implementation of ARQ is illustrated in Figure 7. As illustrated, the first step is to perform initial encoding of the information bits with a mother code. This is normally a low-rate code, called the “mother code”. This mother code is usually a convolutional code, in the case of LTE a Turbo Code is used. In the example, this mother code has a code rate of 1/4. In the next step, the bits that shall be transmitted are obtained from the encoded sequence by either puncturing or repeating it to match the desired code rate. In the example, this is done by puncturing (i.e. selecting only 1/3 of the encoded bits) of the encoded bits. This yields a code rate of 3/4 for the transmitted bits. If the receiver now detects an error in the received data that is not correctable by the FEC, it requests a retransmission

![Figure 7: ARQ with Chase Combining [Dahl08]](image-url)
from the sender. The sender will reply this request by sending exactly the same bits again. The two sets of received symbols will then be combined and thereby the accumulated energy is doubled. This increases the SNR and therefore the chances are better to decode this block successfully. If this fails though, the next retransmission will be requested, which will again increase the accumulated energy.

### 3.1.2 Soft Combining

This behaviour can also be extended in a way that the code rate will be lowered with each retransmitted version of the signal. A procedure called “Soft combining” is illustrated in Figure 8. In the “Soft Combining” procedure, the first transmitted version of bits contains all systematic bits and a limited number of redundancy bits. If this limited number of redundancy bits is not sufficient to correct the transmission errors, the receiver will request the next block. This next version contains other parity bits, thus increasing the amount of redundancy in the receiver’s buffer. With the increased number of parity bits, the receiver tries to decode the block again. If this fails again, another block will be requested until the block has been decoded successfully.

In the example used here, this procedure can only be done three times, since there are no more unsent parity bits in the sender’s buffer after three retransmissions have been send, because the code rate matches the code rate of the mother code. In this case, the transmission will restart at the beginning with the systematic bits and only the amount of accumulated energy in the receiver will increase and therefore hopefully lead to a successful transmission. In LTE this usually will not happen, because it can be supposed that it is unlikely that the channel quality will recover soon after 3 failed transmissions. Therefore this should be managed by link adaption, which will result in the change of the modulation scheme or code rate.

"Figure 8: ARQ with Soft Combining [Dahl08]"

As mentioned, the code rate can be changed by either HARQ or link adaption. The reason why the two methods are used is because of their different nature. HARQ can quickly adapt
to changing conditions, because it automatically decreases the code rate upon reception errors. The advantage of this implicit adaption is the fast reaction. It has a severe drawback though: The delay time of the channel is increased. Therefore, this correction is only used for suddenly occurring transmission errors. If the signal quality decreases for a longer time, the increased delays are not desirable. At this point, link adaption adjusts the code rate or modulation scheme to obtain a lower error rate. This will not affect the delay time of the channel. Of course, the reaction time of the link adaption is long because it needs an explicit call based on a measurement of the signal quality or the error rate.

3.2 Implementation in LTE

In LTE, multiple parallel HARQ processes run in the UE and the eNodeB. A unique process has to run for each subcarrier and for each user in the cell. The packets have to be send with a process-id to identify the HARQ process that this packet belongs to.

Without any feedback about the successful reception of a packet, the sending device has no knowledge if it should send a packet with redundancy bits or continue with the next packet. To transmit this information for the downlink channel, the UE sends a “new-data indicator” to the eNodeB to notify it to clear its buffer and send the next data packet. This indicator is only a single bit and is send through the uplink channel.

3.2.1 Downlink Channel

For the downlink channel, an asynchronous and adaptive version of HARQ is used. Asynchronous means that there is no strict timing for the retransmission. The retransmitted packet is scheduled like any other data packet and the timing is up to the scheduler. It is desirable that this packet has a very high priority to avoid a long delay time. In the adaptive version, the packet does not have to be send via the same subcarrier and also the modulation scheme might be changed to adapt to different channel conditions. Therefore, when sending the redundancy version, the eNodeB is not bound to any limitations concerning subcarrier, modulation scheme, code rate or time interval and can allocate the resources most efficiently to the UEs in the cell.

3.2.2 Uplink Channel

For the uplink channel, synchronous and non-adaptive HARQ is used. In the synchronous form, the retransmission of packets or the additional transmission of redundancy versions is not directed by the scheduler. They will be send after a fixed interval, in LTE standard after 8 packets. The retransmission will use the same resources as the initial transmission, which simplifies the transmission and reduces the need of interaction between eNodeB and UE and therefore reduces the delay of the retransmission. The eNodeB checks the packet for errors and decides whether to request it again. If the packet has to be retransmitted again, the eNodeB sends the notification for this to the UE via the PHICH (Physical Hybrid-ARQ Indicator Channel) or the PDCCH (Physical Downlink Control Channel). Normally the retransmission would be requested via the PHICH where only a single bit has to be send to request the
retransmission from the UE 8 packets later using the same resources as the initial transmission. For special occasions, the eNodeB has the possibility to request a certain redundancy version of the packet or to schedule it to another frequency to avoid collisions. The request for a certain redundancy version is useful if for example the first version with the systematic bits is completely lost. Since in LTE Turbo Codes are used and thus the first bits are the most important bits, it would be more efficient to request the retransmission of these bits than trying to recover them by using only the parity bits from the redundancy blocks. The explicit allocation to a specific subcarrier is necessary to avoid fragmentation in the frequency domain. The SC-FDMA transmission scheme in the uplink channel can only work on adjacent subcarriers because in LTE the “Localised Mode” of SC-FDMA is used\cite{Han09}. Therefore, a fragmentation has to be avoided to efficiently use the available bandwidth and the eNodeB can force the UE to send on a specific subcarrier if frequency fragmentation would occur otherwise.

4 Conclusion

4.1 Scheduling

The most important aspect of scheduling is the fairness towards users because otherwise some users would have problems reasonably making use of the service. Unfairly treated users might suffer from “user starvation” and therefore would not be scheduled for transmission in time and their connections will timeout or suffer from unacceptable delays. Out of the three presented scheduling algorithms, with respect to user fairness, the Proportional Fair Scheduler is the best approach. It still is not perfect and for some exceptions, for example the user constantly coming closer to the eNodeB, special precautions have to be made. Nevertheless it is very likely that the schedulers in LTE implementations will be based on modified versions of the Proportional Fair scheduler.

4.2 HARQ

LTE has a very flexible and powerful system to compensate transmission errors. The channel quality will be constantly measured and the Link Adaption will adapt the modulation scheme used in the channel to this measurement. A Forward Error Correction based on Turbo Code is used to correct a certain amount of errors in each packet. The rate of this code (and therefore the amount of correctable bits) depends on the channel quality and can is set explicitly by punctation of a mother code. If a packet contains to many wrong bits and the FEC fails to decode it, ARQ will request another version of the packet. Due to the used Soft Combining scheme, this packet will increase the redundancy bits in the receivers buffer and therefore also lower the code rate implicitly. This rather complex mechanism of three different ways to adapt the code rate gives a high flexibility to adapt to different signal conditions without wasting too much data rate for redundancy or introducing long delay times.
References


