Journal of Computational Information Systems 8: 2 (2012) 695–707
Available at http://www.Jofcis.com

A Channel Quality-aware Scheduling and Resource Allocation Strategy for Downlink LTE Systems

Shih-Jung WU[*]

Department of Innovative Information and Technology, Tamkang University, I-lan County, R.O.C.

Abstract

Today, the main purpose of a scheduler for Long Term Evolution (LTE) is to provide the best system performance. However, it may decrease the system performance to have latency and starvation of lower priority connections in a resource allocation phase. There has been little research performed on LTE downlink scheduling and resource allocation. This paper proposes an efficient algorithm that includes scheduling strategies and resource allocation mechanisms, to avoid the latency or starvation of lower priority connections and to maintain system performance in downlinks of LTE. The algorithm discusses five levels of bandwidth request situations to assign priority and to allocate the bandwidth for each connection. Therefore, we design an LTE downlink scheduling scheme and a resource allocation strategy that not only aims to achieve the system's highest performance but also avoids latency and starvation problems. As shown in the results of simulations, the proposed algorithm can provide proportional fairness and high system performance in downlinks of LTE systems.

Keywords: LTE; Downlink Scheduling; QoS; CQI; Resource Management

1 Introduction

Long-term Evolution (LTE) is an important technology to transfer from circuit switch networks to All-IP network architectures [1, 2]. LTE has been identified as a new wireless standard by the 3rd Generation Partnership Project (3GPP), which is using the VoIP to transmit voice services and to packet data for all of the services. It could provide the downlink peak rate of 100 Mbps through the OFDMA and SC-FDMA to provide a higher bandwidth, lower latency, and better QoS. The scheduler is an important issue in the MAC layer for system performance. The scheduler in the MAC layer is the main factor that affects the system performance and the resource reusability [3, 4, 10]. In general, designing a scheduler for wireless networks is more difficult and more important than for wired networks because of restrictions on radio resources and variations in channel conditions. The scheduler in LTE aims to maximize system performance. However, it may decrease the system performance for the latency or starvation of connections that have lower priority if the scheduler is only concerned with high throughput. We propose an efficient

[*] Corresponding author.
Email address: wushihjung@mail.tku.edu.tw (Shih-Jung WU).

January 2012

scheduling strategy and resource allocation mechanism to maintain high system performance and to preserve the proportional fairness of the resource allocation.

The proposed algorithm is called a proportional fairness packet scheduling algorithm (PFPS). PFPS restrictively adjusts the priority of the users according to the Channel Quality Indicator (CQI) and allocates the bandwidth according to the variations in the user requests. There are two phases in this algorithm: priority assignment and resource allocation. In the priority assignment phase, the service connections are categorized as real-time (RT) service connections and non-real-time (NRT) service connections. Each category has its own queue to put into the service requests, separately. Otherwise, the emergent queue is used to handle the connections with lower priority that are suffering from latency or starvation. The applicable resource will be allocated according to the upper and lower bound of the current request's bandwidth in the resource allocation phase.

This paper is organized as follows. Related work is shown in Section 2. The proposed algorithm (PFPS) is described in Section 3. The simulation results are presented and discussed in Section 4. Finally, the conclusions are given and future work is described in Section 5.

2 Related Work

Signal processing is divided into voice and data in LTE. The data are transferred and processed on an All-IP network architecture based on a packet switching mechanism. The eNodeB has replaced the Radio Network Controller (RNC) in the WCDMA system [5]. The major wireless transmission technology of LTE is OFDMA. Moreover, the basic architecture of the signal uses OFDM. Multiple Input and Multiple Output (MIMO) [14] could be utilized to improve the transmission performance in LTE. However, in this paper, we do not mention the MIMO issues. OFDMA inherits the advantages of OFDM and improves the multiplex processing control to increase the average transmission rate. OFDMA efficiently arranges the frequency band by using both Time Division Duplexing (TDD) and Frequency Division Duplexing (FDD). FDD utilizes a symmetrical frequency to access the downlink and uplink data transmission. On the other hand, TDD separates the transmitting and receiving channel by time vision. The transmitting and receiving channel use the same frequency in different time slots as the subscriber.

The Channel Quality Indicator (CQI) is the measurement of the channel quality in a wireless network. A higher CQI value usually indicates that the channel has a better channel quality. The CQI of channels can be calculated by the signal-to-noise ratio (SNR), the bit error rate (BER), the signal to interference plus noise ratio (SINR), and the packet loss rate (PLR) [6]. With a 5-bit CQI value (from 0 to 30), a higher CQI with a better channel quality corresponds to a given transport-block size, a modulation scheme, and the number of channelization codes [7].

The main purpose of LTE scheduling aims to provide better resource utilization and channel quality for mobile devices by using a variation in channels. LTE could utilize a variety of channels in the frequency domain and time domain due to the OFDMA architecture. The channel signal would be modulated according to the CQI value of each connection between the mobile device and eNodeB. CQI also selects the appropriate antenna module except for calculating the immediate channel quality in the frequency domain. The MAC layer of LTE is responsible for selecting the size of the block, the modulation [12, 13], and the antenna assignment. The decision of scheduling is based on the TDD mode and then on transferring to the PHY layer. Figure 1 introduces the downlink scheduler in the LTE system.

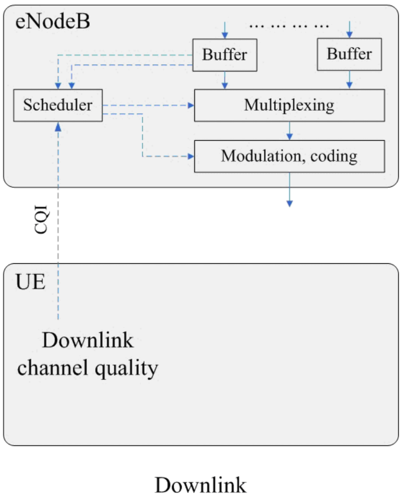

Fig. 1: Downlink scheduler in LTE

We introduce three types of famous scheduling algorithms: maximum rate (Max-rate) [7], round robin (RR)[8], and proportional fair (PF)[9, 11].

Maximum rate (Max-rate): The priority of each user is assigned according to the CQI value to match the objective of the LTE scheduler in Max-rate [7]. The higher CQI would be assigned with a higher priority. Unfortunately, low priority connections will suffer starvation when the total bandwidth cannot satisfy the total requests.

Round robin (RR): Round robin allocates the equivalent time interval to each user [8]. It can maintain fairness for all of the connections and can prevent starvation, but, in doing so, violates the main objective of attaining high system performance for the LTE.

Proportional fair (PF): The PF algorithm is defined with equations (1) and (2) in [9]. This algorithm allocates the resource blocks to users according to a comparison between the theoretical assignment and the actual assignment.

$$P_i(t) = \frac{r_i(t)}{\overline{R}_i(t)} \tag{1}$$

$$P_i(t) = \frac{r_i(t)}{\overline{R}_i(t)^{\beta_i(t)}} \tag{2}$$

Where $P_i(t)$ is the priority for user i at slot t, $r_i(t)$ represents the request data rate, and $\overline{R}_i(t)$ is the average data rate of user i at time slot t. $\beta_i(t)$ indicates the channel with a different data rate. In this paper, we consider every service connection i (not a user), and we calculate the priority according to the ratio of bandwidth request to allocated bandwidth resource in the last frame for its service type. We set up the bandwidth request for RT and NRT service. Less of the bandwidth resource is allocated in the current frame for a service connection i that has a higher priority for being served in the next frame. We utilize this PF method and compare it with our PFPS in simulation.

3 The Downlink Scheduling Scheme

The main objective of this paper aims to design a scheduling algorithm that is adopted by the LTE standard. Each user is allocated the requested resource according to the predefined QoS parameters. This paper proposes the proportional fairness packet scheduling strategy for downlink LTE (PFPS). PFPS is divided into two parts: priority assignment and resource allocation. The proposed PFPS is the scheduling algorithm that is designed with the TDD mode and a centralized architecture. As shown in Figure 2, PFPS is a frame-based scheduling algorithm. Each frame is organized by 10 subframes [7]. PFPS accesses the scheduling procedure before the end of each frame and finishes the scheduling task before the next frame begins.

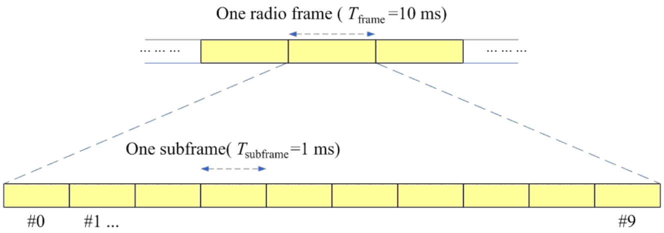
One radio frame (T_{frame}=10 ms)
One subframe($T_{subframe}$=1 ms)
#0
#1 ...
#9

Fig. 2: Frame structure

3.1 Proportional fairness packet scheduling architecture

The PFPS is the priority-based scheduling algorithm that indicates the transmission ranking by the assigned priority to each connection. PFPS improves the situation of losing guaranteed QoS by dynamically adjusting the priority of the user demands. The service types are categorized into two types: real-time service connections (RT) and non-real-time service connections (NRT). The RT service connections will be served first. Additionally, every user (user equipment) can have many service connections. As shown in figure 3, the priority assignment could be divided into two parts: CQI ranking and fairness control. CQI ranking is decided by the CQI value of each connection to guarantee the whole system performance by satisfying the requests of all of the users. Fairness control promotes the priority level of the lower priority connections to avoid any service interrupts. The bandwidth requirement allocation distributes resources according to the total bandwidth and the demanded upper bound and lower bound bandwidth. We need a setup minimum bandwidth request b_{min-RT}, $b_{min-NRT}$ to evaluate the starvation and latency and maximum bandwidth request b_{max-RT}, and $b_{max-NRT}$ for any service connection. The minimum and maximum bandwidth request for any service connection also helps us to estimate the upper bound and the lower bound for the total bandwidth request.

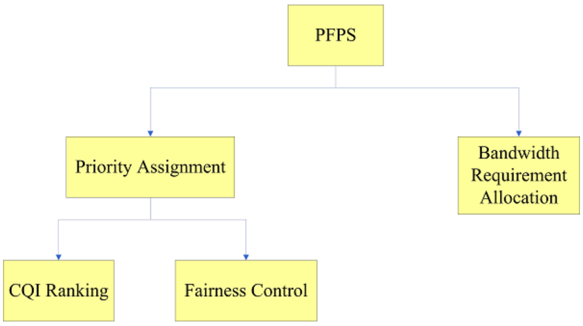

Fig. 3: PFPS architecture

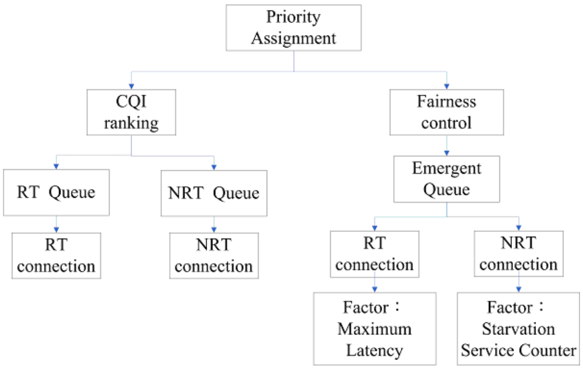

Fig. 4: Priority assignment

3.2 Scheduling

Figure 4 is the priority assignment architecture. The emergent queue is involved in satisfying the QoS and in preventing service interruption of lower service priority in fairness control. The maximum latency and starvation service counters are used to handle the priority promotion of RT services and NRT services separately.

Because RT services emphasize the latency problem, the priority assignment of RT services focuses on the degree of latency when fitting in the requests of QoS. The packet will be put into the emergent queue when it satisfies equation (3). Equation (3) shows that the tolerable waiting time is shorter than the length of one frame. The packet will be put into the emergent queue because the packet needs to be delivered in the next frame to satisfy the QoS.

$$\zeta_i - (T_c - T_i^a(j)) \le T_{frame} \qquad \forall i = 1 \cdots N_{RT}, \forall i \in \Omega_{RT} \tag{3}$$

ζ_i represents the maximum latency of connection i. T_c indicates the system current time. $T_i^a(j)$ is the arrival time of the jth packet in connection i. T_{frame} is the length of one frame. N_{RT} is the number of RT services in a downlink. Ω_{RT} shows the set of RT services in a downlink. In other words, if the serving time of the RT service exceeds one frame duration, it will cause latency.

The starvation service counter is used to detect the occurrences of starvation in the NRT services. The counter will be incremented by one when the transmission rate is 0 in the last frame. The starvation of the connection is defined as the value of the counter exceeding the threshold η. The connection will be put into the emergent queue to avoid starvation. Equation (4) indicates that the connection i of the m_{th} frame has not been served or the allocated bandwidth resource is less than the minimum bandwidth request for NRT in the $(m-1)$-th frame if equation (4) is satisfied. At the same time, the starvation service counter of connection i will increase by 1. Otherwise, the connection i will be put into an emergent queue if it satisfies equation (5), in which case the service interrupt of connection i exceeds the tolerable quantity. The term $b_i^a(m-1)$ represents the allocated bandwidth resource of connection i in frame $m-1$, and $\phi_i(m)$ is the starvation service counter value of connection i in frame m. The set of NRT services in downlink is shown as Ω_{NRT}, and N_{RT} indicates the number of NRT services in downlink.

$$b_i^a(m-1) \le b_{min-NRT} \qquad \forall i \cdots N_{NRT}, \forall i \in \Omega_{NRT} \tag{4}$$

$$\phi_i(m) \ge \eta \qquad \forall i \cdots N_{NRT}, \forall i \in \Omega_{NRT} \tag{5}$$

There are three queues that are used in this paper; these queues are the RT Queue (RT_Q), the NRT Queue (NRT_Q), and the Emergent Queue (E_Q). The RT_Q and the NRT_Q are used separately to hold the ranked packets of RT and NRT service connections. E_Q is used to hold the packets that have exceeded the maximum latency or the starvation service counter. The packets are ranked by the CQI value, which is divided into 31 levels. A higher CQI value indicates a higher priority. In RT services, we first check whether or not the RT services exceed the maximum latency. If the RT services do not exceed the maximum latency, the RT services are put into the RT_Q according to the priority value, which is assigned based on the CQI value. Otherwise, the RT services will be put into the E_Q. Figure 5 represents the flowchart of the RT services determination.

In NRT services, we first check whether or not the NRT services exceed the starvation service counter threshold η. If the RT services do not exceed the threshold η, then the NRT services

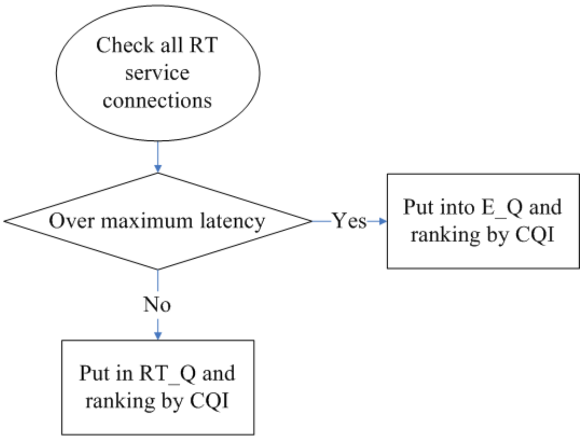
Check all RT service connections
Over maximum latency
Yes
Put into E_Q and ranking by CQI
No
Put in RT_Q and ranking by CQI

Fig. 5: RT service determination flowchart

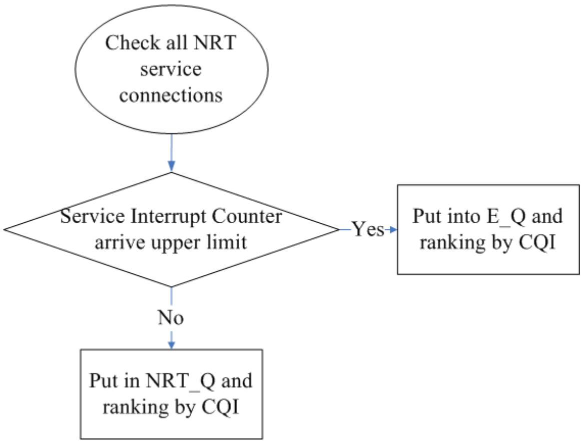
Check all NRT service connections
Service Interrupt Counter arrive upper limit
Yes
Put into E_Q and ranking by CQI
No
Put in NRT_Q and ranking by CQI

Fig. 6: NRT service determination flowchart

are put into the NRT_Q according to the priority value that is assigned based on the CQI value. Otherwise, the NRT services will be put into the E_Q. Figure 6 represents the flowchart of NRT services determination.

The emergent queue is used to hold the packets that exceed the maximum latency or the starvation service counter threshold. The RT services have a higher priority than the NRT services in E_Q. The RT services and the NRT services are ranked according to their own priority separately, as shown in figure 7.

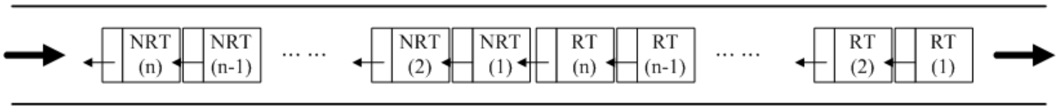

Fig. 7: Emergent queue

3.3 Resource allocation

The resource allocation is processed according to the results of section III-B. There are five cases in resource allocation: Case I — the total bandwidth (B) is less than the total minimum request bandwidth of the RT services (RT_min); Case II — the total bandwidth (B) is equal or more than the total minimum request bandwidth of the RT services (RT_min); Case III — the total bandwidth (B) is equal to or more than the total minimum request bandwidth of the RT services (RT_min) and the NRT services (NRT_min); Case IV — the total bandwidth (B) is equal to or more than the total maximum request bandwidth of the RT services (RT_max) and the total minimum request bandwidth of the NRT services (NRT_min); and Case V — the total bandwidth (B) is equal to or more than the total maximum request bandwidth of the RT services (RT_max) and NRT services (NRT_max). Figure 8 shows the architecture of the resource allocation.

In Case I, $B <$ RT_min. First, we check whether or not the E_Q is empty. If the E_Q is empty, then we allocate the minimum request bandwidth to each service connection in the E_Q. Next, we allocate the remaining bandwidth according to the priorities until the remaining bandwidth is empty with respect to the RT services in the RT_Q with the maximum request bandwidth. Otherwise, if the E_Q is empty, we allocate the maximum request bandwidth according to the

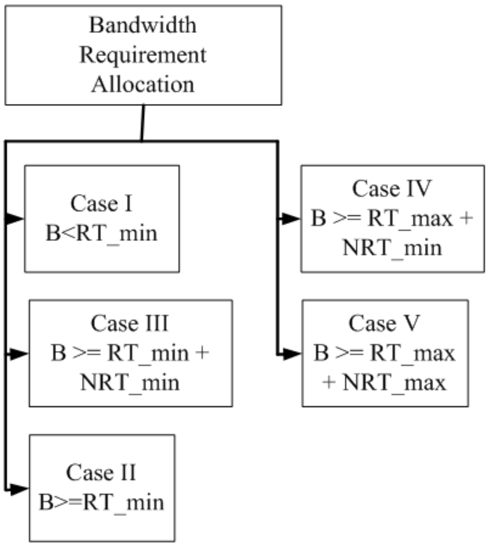
Bandwidth Requirement Allocation
Case I B<RT_min
Case IV B >= RT_max + NRT_min
Case III B >= RT_min + NRT_min
Case V B >= RT_max + NRT_max
Case II B>=RT_min

Fig. 8: Resource allocation

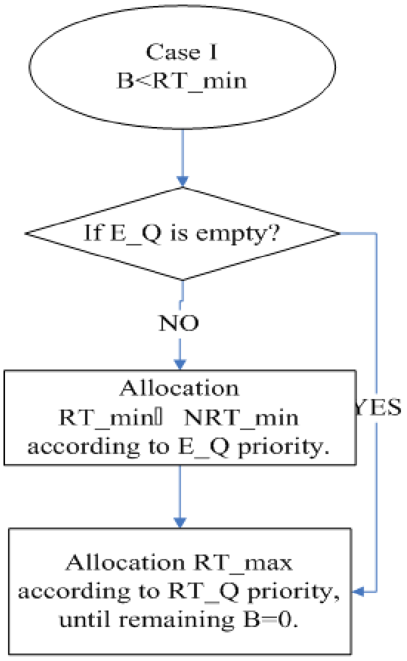
Case I B<RT_min
If E_Q is empty?
NO
YES
Allocation RT_min、NRT_min according to E_Q priority.
Allocation RT_max according to RT_Q priority, until remaining B=0.

Fig. 9: Flowchart of Case I

priorities until the remaining bandwidth is empty to the RT services in the RT_Q. These actions are shown in figure 9.

In Case II, B >= RT_min. First, we check whether or not the E_Q is empty. If the E_Q is not empty, then we allocate the minimum request bandwidth of each connection in E_Q. Next, we allocate the minimum request bandwidth to RT services in RT_Q if the remaining bandwidth is more than the total minimum request for bandwidth of RT services in RT_Q and allocate the maximum request bandwidth to NRT services in NRT_Q. The next step is to allocate the remaining bandwidth to RT services until there is a match to the maximum request bandwidth according to the priority. Otherwise, if the E_Q is empty, we do not care about the E_Q, and we execute the above-mentioned steps directly. These actions are shown in figure 10.

In Case III, B >= RT_min + NRT_min. Check whether or not the E_Q is empty. If the E_Q is not empty, then we allocate the minimum requested bandwidth of each connection in E_Q, we allocate the minimum requested bandwidth of RT services in RT_Q, we allocate the minimum requested bandwidth to NRT services in NRT_Q, and we allocate the remaining bandwidth to RT services until there is a match to the maximum requested bandwidth according to the priorities when the remaining bandwidth is empty. Otherwise, if the E_Q is empty, then we allocate the minimum requested bandwidth to RT services in RT_Q, we allocate the minimum requested bandwidth to NRT services in NRT_Q, and we allocate the remaining bandwidth to RT services until we match the maximum requested bandwidth according to the priority until the remaining bandwidth is empty. The flowchart is shown in figure 11. In this case, system will never make the latency on RT services or starvation on NRT services because the available bandwidth is more than the request.

In Case IV, B >= RT_max + NRT_min. First, we allocate the maximum requested bandwidth to RT services according to its priority in RT_Q. Then, we allocate the minimum requested

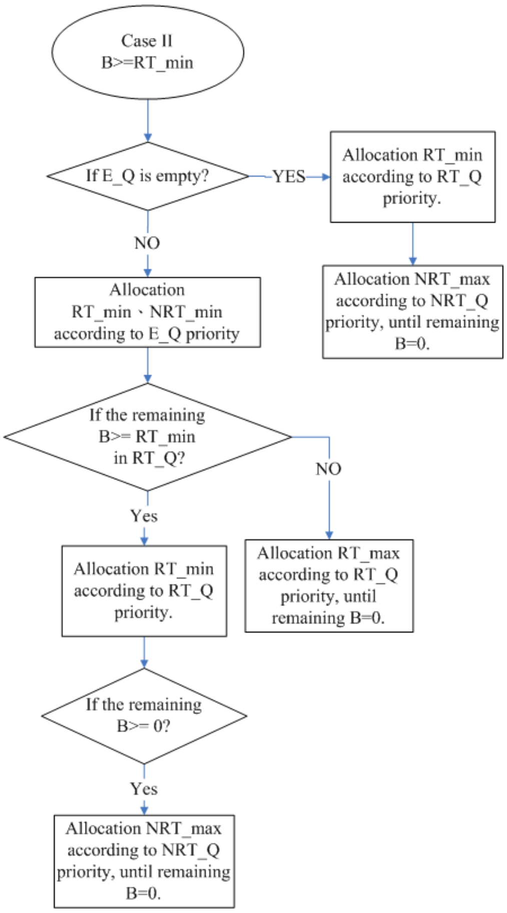
Case II
B>=RT_min
If E_Q is empty?
YES
Allocation RT_min according to RT_Q priority.
Allocation NRT_max according to NRT_Q priority, until remaining B=0.
NO
Allocation RT_min、NRT_min according to E_Q priority
If the remaining B>= RT_min in RT_Q?
NO
Allocation RT_max according to RT_Q priority, until remaining B=0.
Yes
Allocation RT_min according to RT_Q priority.
If the remaining B>= 0?
Yes
Allocation NRT_max according to NRT_Q priority, until remaining B=0.

Fig. 10: Flowchart of Case II

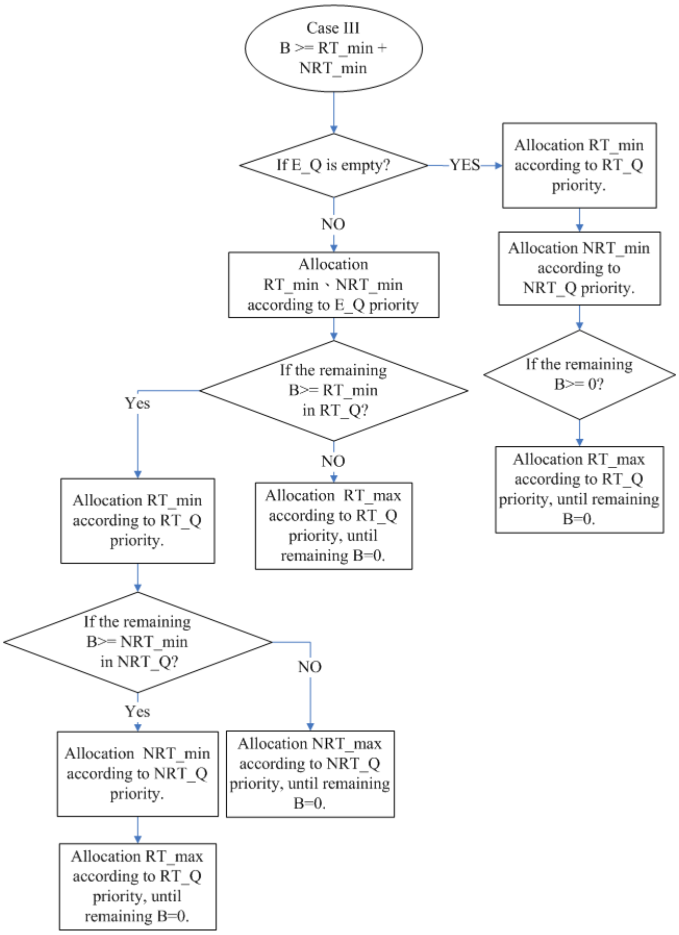
Case III
B >= RT_min + NRT_min
If E_Q is empty?
YES
Allocation RT_min according to RT_Q priority.
Allocation NRT_min according to NRT_Q priority.
If the remaining B>= 0?
Allocation RT_max according to RT_Q priority, until remaining B=0.
NO
Allocation RT_min、NRT_min according to E_Q priority
If the remaining B>= RT_min in RT_Q?
Yes
Allocation RT_min according to RT_Q priority.
NO
Allocation RT_max according to RT_Q priority, until remaining B=0.
If the remaining B>= NRT_min in NRT_Q?
Yes
Allocation NRT_min according to NRT_Q priority.
NO
Allocation NRT_max according to NRT_Q priority, until remaining B=0.
Allocation RT_max according to RT_Q priority, until remaining B=0.

Fig. 11: Flowchart of Case III

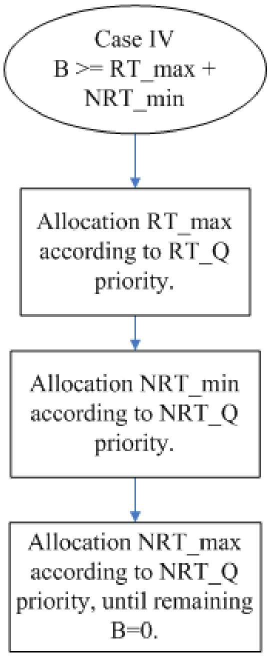
Case IV
B >= RT_max + NRT_min
Allocation RT_max according to RT_Q priority.
Allocation NRT_min according to NRT_Q priority.
Allocation NRT_max according to NRT_Q priority, until remaining B=0.

Fig. 12: Flowchart of Case IV

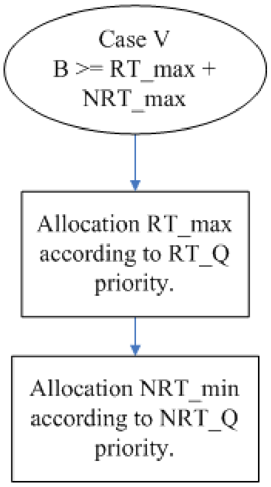
Case V
B >= RT_max + NRT_max
Allocation RT_max according to RT_Q priority.
Allocation NRT_min according to NRT_Q priority.

Fig. 13: Flowchart of Case V

bandwidth to NRT services according to the priority in NRT_Q. Finally, we allocate the remaining bandwidth to NRT services until we match the maximum requested bandwidth according to the priority until the remaining bandwidth is empty. These actions are shown in figure 12.

In Case V, B >= RT_max + NRT_max. We allocate the maximum requested bandwidth to RT services according to the priority in RT_Q. Then, we allocate the maximum requested bandwidth to NRT service connections according to its priority in NRT_Q. A flowchart is shown in figure 13.

4 Simulation Results and Analysis

A simulation was used to compare the three existing methods from the literature: max-rate, round robin (RR), and proportional fairness (PF). The assumptions and the parameters are described as follows:

(1) Description of simulation assumptions:

- TDD-based network architecture
- Scheduling decision is operated in the BS side.
- Assume that all of the connections are created after the call admission control (CAC).
- The number of connections is fixed.
- Every user (mobile station) maybe has many service connections.

(2) Simulation model is shown in figure 14.

(3) Simulation parameter descriptions are given in Table 1:

The simulations are discussed on the latency and starvation for five different bandwidth requests. We observed the changes in bandwidth allocation.

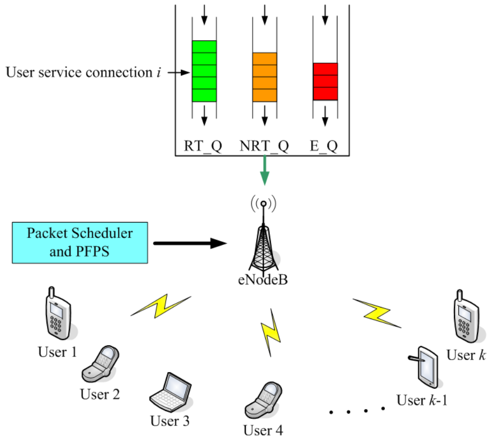

Fig. 14: Simulation model

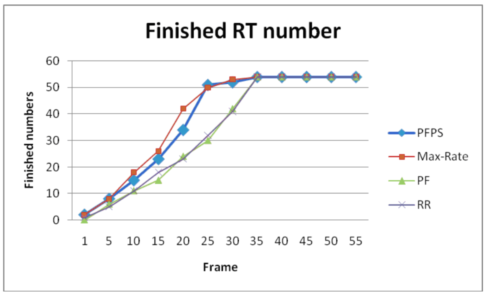

Fig. 15: For the RT service connections in simulation time

Table 1: Simulation parameters

RT service connections	0∼100
NRT service connections	0∼100
Total amount of bandwidth	20Mbps
Frame duration	10ms
Simulation time	100 frames
Starvation threshold	5 (50ms)
Request of RT_min per frame (b_{min-RT})	1000 Byte
Request of RT_max per frame (b_{max-RT})	1200 Byte
Request of NRT_min per frame ($b_{min-NRT}$)	500 Byte
Request of NRT_max per frame ($b_{max-NRT}$)	700 Byte

4.1 System performance

Figure 15 shows the average system performance of the RT services. Max-rate obtains the highest system performance because it has the highest throughput. RR and PF cannot efficiently improve system performance because of their consideration of fairness first. The proposed PFPS can efficiently solve the problem of the latency and starvation of Max-rate and produce system performance that is better than the RR and PF.

4.2 Latency

4.2.1 B < RT_min (Case I)

Figure 16 indicates the latency of RT services in Case I. RR and PF suffers a higher amount of latency of RT services than Max-rate and PFPS. This effect is caused by implementing fairness for all of the services. Max-rate has a higher amount of latency of RT services than PFPS because Max-rate cares only about the throughput.

4.2.2 B >= RT_min (Case II), B >= RT_min + NRT_min (Case III), B >= RT_max + NRT_min (Case IV)

Figure 17 represents the latency of RT services in Case II, Case III, and Case IV. The Max-rate and PFPS solve the latency problem of RT services in these three cases because they consider the RT services first. In contrast, the RR and PF still suffer the latency problem of RT services because they consider fairness.

4.2.3 B >= RT_max + NRT_max (Case V)

In figure 18, the latency of RT services in Case V is discussed. Because the available bandwidth is more than the requested bandwith, all of the scheduling algorithms can address the latency problem of the RT services.

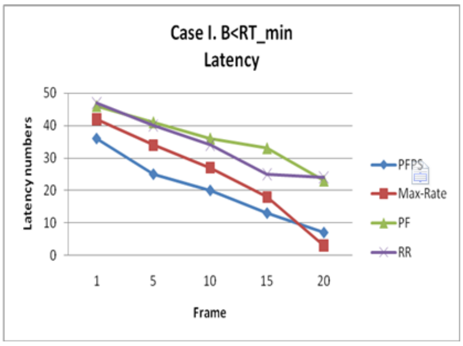

Fig. 16: Latency numbers (Case I)

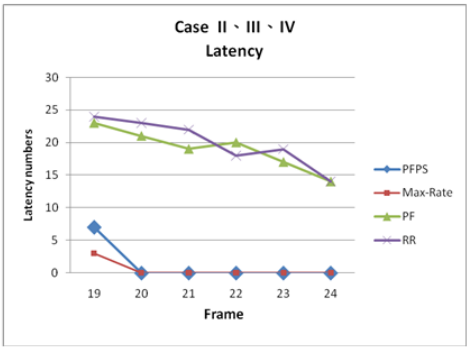

Fig. 17: Latency numbers (Case II, III, IV)

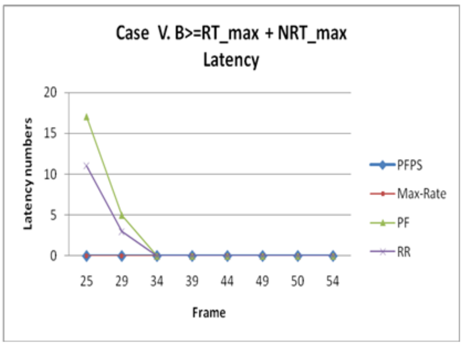

Fig. 18: Latency numbers (Case V)

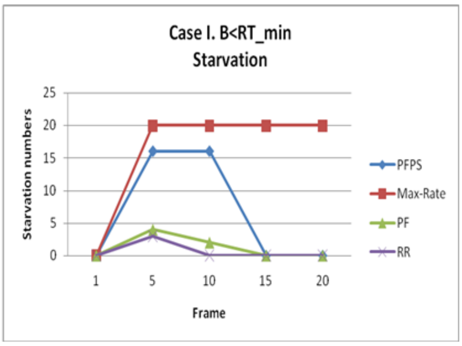

Fig. 19: Starvation numbers (Case I)

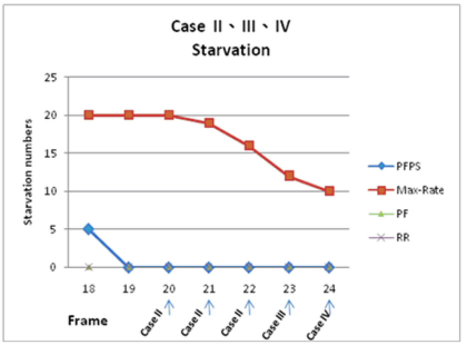

Fig. 20: Starvation numbers (Case II, III, IV)

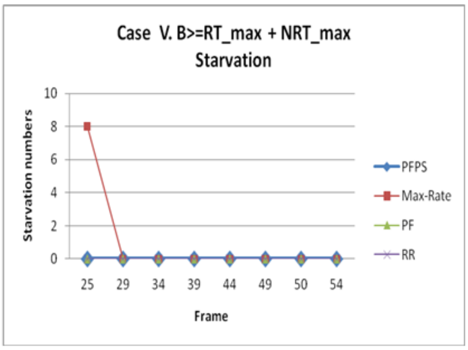

Fig. 21: Starvation numbers (Case V)

According to these three simulation results, we could conclude that the PFPS could efficiently solve the latency problem of RT services and improve the system performance at the same time.

4.3 Starvation

4.3.1 B < RT_min (Case I)

Figure 19 indicates the starvation of NRT services in Case I. RR and PF have a lower number of starvation services than Max-rate and PFPS because they consider fairness first. For the Max-rate, it suffers a high starvation service number because it considers only the throughput. The proposed PFPS can address the starvation problem of the NRT service by using the starvation service counter.

4.3.2 B >= RT_min (Case II), B >= RT_min + NRT_min(Case III), B >= RT_max + NRT_min (Case IV)

Figure 20 represents the starvation of RT services in Case II, Case III, and Case IV. As we can see, RR and PF do not have the starvation problem issue here. The proposed PFPS can also solve the starvation problem in these three cases by using the starvation service counter. For the Max-rate, it will still suffer the starvation problem due to having concerns only about improving the throughput.

4.3.3 B >= RT_max + NRT_max (Case V)

In figure 21, because the available bandwidth is more than the request, all of the scheduling algorithms can address the starvation problem of the RT services.

5 Conclusion

The efficient wireless resource management and the scheduling algorithm can improve the system performance and meet the QoS request of each user. The design of the scheduler in LTE has to consider the limitations of the wireless resources and the variations in the channel quality. The system performance may decrease due to latency or starvation of lower priority services. In this paper, we propose the PFPS algorithm to maintain the fairness of all of the services and avoid latency or starvation. As shown in the simulation results, PFPS has a higher throughput than RR and PF. Meanwhile, it has more fairness than Max-rate. In the future, we will consider designing the uplink scheduling algorithm in LTE. Moreover, the complete scheduler will be created based on the proposed uplink and downlink scheduling algorithms.

References

[1] ITU Telecommunications indicators update 2006, http://www.itu.int/ITU-D/ict/statistics/.

[2] In-stat Report. Paxton. *The broadband boom continues: Worldwide subscribers pass 200 million*, No. IN0603199MBS, March 2006.

[3] Hossam Fattah, and Cyril Leung, An overview of scheduling algorithms in wireless multimedia networks, *IEEE Wireless Communications*, vol. 9, no. 5, pp. 76 – 83, Oct. 2002.

[4] Yaxin CAC and Victor O. K. Li, Scheduling Algorithms in Broad-Band Wireless Networks, *IEEE Proceedings of The IEEE*, vol. 89, no. 1, pp. 76 – 87, Jan. 2001.

[5] E-UTRAN Architecture description, 3GPP TS 36.401, 3GPP specifications [online].: http://www.3gpp.orgj.

[6] A. M. Mourad, L. Brunel, A. Okazaki, and U. Salim, Channel Quality Indicator Estimation for OFDMA Systems in the Downlink, *IEEE 65th Vehicular Technology Conference*, pp. 1771 – 1775, April 2007.

[7] E. Dahlman, S. Parkvall, J. Skold, and P. Beming, *3G Evolution: HSPA and LTE for Mobile Broadband*, First ed.: Elsevier Ltd. p. 176, 2007.

[8] *E-UTRA Downlink User Throughput and Spectrum Efficiency*, Ericsson, Tdoc R1-061381, 3GPP TSG-RAN WG1, Shanghai, China, May 8 – 12, 2006.

[9] A. Jalali, R. Padovani, and R. Pankaj. Data Throughput of CDMAHDR: a High Efficiency-High Data Rate Personal Communication Wireless System, *Proceeding of IEEE Vehicular Technology Conference* (VTC Sprint 2000), vol. 3, pp. 1854 – 1858, May 2000.

[10] Ramli H. A. M., Sandrasegaran K., Basukala R., Leijia Wu, Modeling and simulation of packet scheduling in the downlink long term evolution system, *15th Asia-Pacific Conference on Communications*, Oct. 2009, pp. 68 – 71.

[11] A. Gyasi-Agyei and S. -L. Kim, Comparison of Opportunistic Scheduling Policies in Time-Slotted AMC Wireless Networks, *in 1st International Symposium on Wireless Pervasive Computing*, 2006.

[12] Lin CHEN, Xuelong HU, Research on PAPR Reduction in OFDM Systems, Journal of Computational Information Systems, vol. 6, no. 12, pp. 3919 – 3927, 2010.

[13] Lin CHEN, Xuelong HU, Improved SLM Techniques for PAPR Reduction in OFDM System, Journal of Computational Information Systems, vol. 6, no. 13, pp. 4427 – 4433, 2010.

[14] Dun CAO, Hongwei DU, Ming FU, Cubic Hermite Interpolation-based Channel Estimator for MIMO-OFDM, Journal of Computational Information Systems, vol. 6, no. 14, pp. 4699 – 4704, 2010.