Three-step Iterative Scheduler for QoS Provisioning to Users Running Multiple Services in Parallel

Ankit Bhamri*†, Navid Nikaein*, Florian Kaltenberger*, Jyri Hämäläinen†, Raymond Knopp*
*Eurecom, France (email : firstname.lastname@eurecom.fr)
†Aalto University School of Electrical Engineering, Finland (email : firstname.lastname@aalto.fi)

***Abstract*—Wireless networks are evolving continuously and expected to provide seamless experience for multiple real-time internet applications. Quality-of-service is one of the major component associated with user experience. In this paper, we have considered the provisioning of desired QoS to mobile users that are capable of running multiple internet applications in parallel. For this purpose, a three-step iterative downlink scheduler is proposed for resource management at per-user-per-service level. The scheduler performs sorting in multiple iterations on the basis of three weights. In the first iteration, the scheduler performs sorting based on the throughput weight. The second iteration latency weight and followed by buffer weight in the third iteration. The allocation of resources is done to satisfy the promised QoS to all the services of every user. A comparison is carried out with traditional scheduling algorithms in terms of system throughput, fairness index and percentage of satisfied guaranteed bit-rate users. Results show that the proposed algorithm outperforms existing schemes and the performance is more closer to theoretical system throughput.**

***Keywords*: Iterative-scheduler, Quality-of-service (QoS), Resource management, Real-time services, Traffic modeling**

I. Introduction

Mobile phones now account for almost a fifth of global web usage, as smartphones have become ubiquitous in many parts of the world. People around the globe are accessing internet applications more than ever on their mobile devices with an increase of six percent between 2012-13 [1]. As a result, the wireless network traffic has significantly increased in volume. All measurements in current mobile networks and all forecasts indicate fast increase in mobile data traffic. For example, widely referenced Cisco VNI forecast reports that mobile data traffic grew 70% in 2012 and global mobile data traffic is expected to increase 13-fold between 2012 and 2017 [2]. In addition, the varied internet applications that are accessed on mobile devices have their own requirements in terms of QoS which must be satisfied by the network operators. It is a non-trivial task to manage limited resources for providing promised QoS to all users and efficient scheduling algorithms are required. Conventional scheduling algorithms would result in sub-optimal performance in modern wireless networks [3], [4], [5].

In this paper, we propose a MAC-layer scheduler for efficient resource management in downlink. It is a three-step iterative scheduler that takes into account three crucial parameters for sorting and allocation of resources. The scheduling is done on per-user-per-service basis meaning that it analyzes the parameters associated with all the logical channels (service radio bearers) for every user and apply scheduling algorithm to the packets of logical channels individually rather than just at the user level. The three weights utilized for this iterative algorithm are throughput weight, latency weight and buffer weight. In contemporary frameworks, most of the schedulers allocate resources on user basis, user belonging to a particular service class. However in modern networks, the user can demand several services belonging to different QoS classes. Therefore we treat allocation in two dimension matrix with each block belonging to a particular service of a user. We proposed the technique of two-dimensional buffer management in our recently submitted work [6]. Fig. 1 shows the two-dimensional buffer of *users* $\times$ *logical channels* that needs to be sorted optimally for efficient resource allocation. These logical channels correspond to different services with respective QoS requirements. In this paper, we further utilize this technique by developing an iterative scheduling algorithm for provisioning desired QoS to all the services of every user.

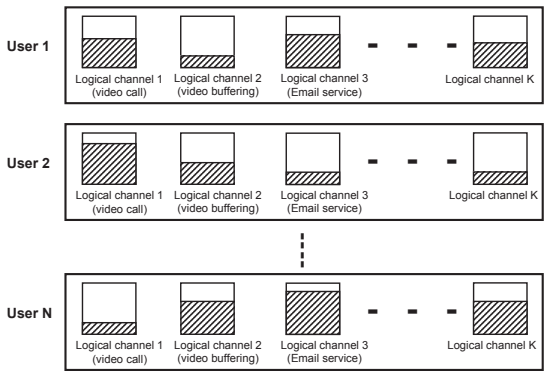

Fig. 1. Two-dimensional buffer

The rest of the paper is organized as follows: Section II defines the required parameters and system setting. In Section III, we describe the scheduler characteristics and algorithm and Section IV provides comparative analysis to demonstrate the gains in terms of total system throughput, percentage of satisfied users and fairness index. The paper is concluded in Section V.

II. System Description

We are considering resource management in dowlink transmission from the base station to the users. The scheduler is designed for all-IP packet switched networks such as 3GPP LTE. In such networks, the buffer of a logical channel consists of packets and we apply scheduling algorithm to the packets of a logical channel for every user in the system. Our analysis considers downlink transmission to N users having K logical channels. Every logical channel is associated with a service belonging to specific QoS class. Depending up on the service requests, there is a buffer queue in the respective logical channel of a user. For our analysis, we consider users requesting multiple services at the same time. Therefore, there can be multiple buffer queues waiting to be scheduled. Each buffer queue is identified by a set of parameters that are utilized for designing an optimal scheduling algorithm. The parameters are categorized in four different groups depending up on their association.

A. Packet-level parameters

Packets are the data units that constitute the logical channels. Every packet has a set of pre-defined parameters as follows:

- Average packet size (APS): Packet size is dependent up on the type of service being served such as ftp packets, browsing, etc. For our analysis we have considered an average packet size for each kind of service.
- Packets inter-arrival time (PIT): This gives the frequency of packets arrival in the buffer queue which is also dependent up on the type of service.
- Packet arrival time (PAT): It is the time-stamp of when the packet arrived in the buffer queue.
- Packet maximum-allowable delay (PMD): This is a pre-defined value for each service type. It is also referred to as the latency constraint.
- Packet remaining time (PRT): This is the time remaining before the packet will be dropped from the buffer queue. It is dependent up on the packet arrival time, packet maximum-allowable delay and current time.

B. Logical channel-level parameters

These parameters characterizes the logical channel/service of a user.

- Number of protocol data units (NPDU): Every logical channel is composed of a number of packets referred to as protocol data units (PDU).
- Head-of-line delay (HOD): It is the time in buffer for the first packet in the buffer queue of a logical channel.
- Buffer size of a logical channel (BS): This is the sum of the all the packet's size in a logical channel.
- Guaranteed bit-rate (GBR):This is the minimum required bit-rate for each service type. There are few services that have non-guaranteed bit-rate.
- Level of traffic (TL): This is factor for traffic modeling of a given service. It varies from 0 to 1 where 0 means there is no traffic and 1 indicates continuous flow of traffic.

C. User-level parameters

The parameters associated with user are:

- Total buffer size (TBS): The sum of the buffer sizes for all logical channels of a user.
- Number of logical channels (NLC): It is the total number of logical channels or services of a user.
- Channel quality indicator (CQI): This is the channel quality feedback received from physical layer. It is an important factor for all channel aware schedulers.
- Subscription type (ST): When a user subscribed to a network operator, it has the option of selecting one of the many subscriptions. Usually, the better the subscription type, better is the promised data-rate. We have defined three subscription types in our analysis: basic, silver and gold.

D. System-level parameters

These are the global parameters of a system.

- Number of active users (NU): It indicates the number of users that are active for transmission/reception in that particular transmission time.
- Maximum allowed scheduled users (MAU): Every system has a limitation on the number of user that can be actually served. This number is usually dependent up on the system bandwidth.
- Frame configuration type (CT): In our analysis, we have used the frequency division duplex (FDD) configuration type.
- Transmission time interval (TTI): This is the smallest unit of transmission time. For example, in LTE, subframe is the TTI.
- Minimum resource allocation unit (MRU): This is the minimum unit of resources in frequency domain that can be allocated for scheduling.

III. Three-step Iterative Scheduling Algorithm

In this section, we describe the three-step iterative MAC-layer scheduler for optimal allocation of resources and providing desired QoS to all the service requested by every user. The scheduler is characterized as follows:

1) **Conversion**: It is based on two-dimensional buffer management composed of users and their logical channels. The two dimensional matrix of *users* $\times$ *logical channels* is converted to single dimension of *blocks*. These blocks are then sorted based on a number of factors for optimal performance. As shown in [6], this step results in increase in the performance gain by applying higher resolution of sorting to logical channels within users. Fig. 2 shows the conversion into single dimension.
2) **Sorting**: Sorting is iteratively done in three steps based on three calculated weights:
   a) *Throughput weight*: To begin with, the blocks are sorted in increasing order of throughput weight given by Equation 1. This is a channel quality aware scheduler, that dynamically sorts the blocks on the basis of

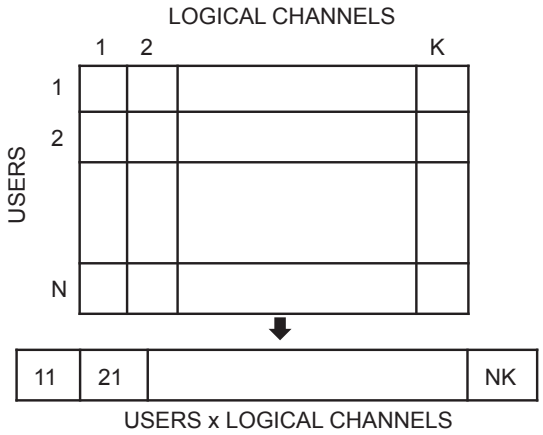

Fig. 2. Transform from 2-dimensional to 1-dimensional system

instantaneous quality of channel that is fed back by the user. This would result in better system throughput

$$\mathrm{TGPT}_{\mathrm{weight}} = \frac{BMT_j}{MPT} \tag{1}$$

where BMT_j is the maximum possible throughput for a block with index j. It is calculated based on the instantaneous CQI of that block and the maximum number of MRU. MPT is the maximum possible throughput of the system depending up on maximum possible CQI and maximum number of MRU. However, there are going to be a number of blocks having same throughput weight and therefore it would need further sorting based on other weights.

b) *Latency weight*: The blocks with same throughput weight are sorted in increasing order of latency weight given by Equation 2. It is the ratio of the packet remaining time and the packet maximum allowable delay for the first packet in queue of every block. This weight gives preference to services that are first in line to be dropped from the buffer queue and as a result adds fairness to the scheduler.

$$\mathrm{Latency}_{\mathrm{weight}} = \frac{PRT}{PMD} \tag{2}$$

c) *Buffer weight*: The blocks with same throughput weight and latency weight are sorted on the basis of buffer weight given by Equation 3. This prevents starvation by prioritizing blocks with higher ratio of buffer weight meaning that the blocks with a long buffer queue are preferable. Buffer weight is dependent up on the buffer size of that block and the total buffer size of the system.

$$\mathrm{Buffer}_{\mathrm{weight}} = \frac{BS_j}{TBS \times NU} \tag{3}$$

3) **Allocation**: Once the blocks are sorted, the resources are allocated to the blocks. The number of resources that are allocated to each block is given by the QoS requirement and buffer size. The blocks with higher QoS requirement are allocated more number of resources in order to satisfy the GBR. Therefore this scheduler can be described as QoS-aware iterative scheduler.

A. Scheduler Algorithm

In this section, we describe the scheduler algorithm in three parts by Algorithm 1, 2 and 3. Algorithm 1 performs the conversion from two-dimension to single dimension. It is followed by sorting in Algorithm 2 and Algorithm 3 shows the steps for allocation of resources to sorted blocks. In Algorithm 3, we calculate the optimal number of required resources for a block on the basis of GBR and the buffer size (BS) of that block.

Algorithm 1 Convert *users* $\times$ *logical channels* into *blocks*

```
lc_count = 1
block_count = 1
while lc_count ≤ NLC_b do
  user_count = 1
  while user_count ≤ NU_s do
    if BS_b > 0 then
      block[block_count] = [user_count, lc_count]
      block_count = block_count + 1
    end if
    user_count = user_count + 1
  end while
  lc_count = lc_count + 1
end while
```

IV. Comparative Analysis

In this section, we compare the performance of our proposed scheduler with the traditional schedulers in terms of system throughput, fairness index and percentage of satisfied users and their services. Our analysis is limited to the gain coming from MAC-layer scheduler, therefore we assume that all the scheduled packets are received and decoded at the physical layer successfully.

A. Simulation Setup

This scheduler can be applied to all the wireless standards with all-IP network, therefore for this comparative analysis we consider 3GPP LTE setting. Table 1 shows the simulation parameters used for the comparative analysis.

Parameters	Values
Transmission Bandwidth	5MHz (25 RBs)
Simulation Window	1000 LTE frames
Logical Channels	9
QoS Class Support	GBR and Non-GBR
Traffic Model	Random

TABLE I
Simulation Parameters

Algorithm 2 Sort *blocks*

```
count1 = 0
while count1 ≤ total_block_count do
  count2 = count1 + 1
  while count2 ≤ total_block_count do
    if TGPT_weight[count2] > TGPT_weight[count1] then
      swap(count1, count2)
    else
      if TGPT_weight[count2] == TGPT_weight[count1] then
        if Latency_weight[count2] > Latency_weight[count1] then
          swap(count1, count2)
        else
          if Latency_weight[count2] == Latency_weight[count1] then
            if Buffer_weight[count2] > Buffer_weight[count1] then
              swap(count1, count2)
            end if
          end if
        end if
      end if
    end if
    count2 = count2 + 1
  end while
  count1 = count1 + 1
end while
```

Algorithm 3 Calculate *alloc_resources_to_block*

```
while number_of_resources_remaining > 0 do
  while number_of_blocks_scheduled ≤ max_allowed_blocks do
    if number_of_users_scheduled ≤ max_allowed_sched_user then
      req_resource_alloc_units = calculate(GBR[j], BS[j])
      allocated_resources_to_block[j] = req_resource_alloc_units + allocated_resources_to_block[j]
      if user_scheduled[n] ≠ 1 then
        user_scheduled[n] = 1
      end if
      if block_scheduled[j] ≠ 1 then
        block_scheduled[j] = 1
      end if
      number_of_users_scheduled = number_of_users_scheduled + 1
      number_of_blocks_scheduled = number_of_blocks_scheduled + 1
      number_of_resources_remaining = number_of_resources_remaining + min_resource_alloc_unit
    end if
  end while
end while
```

In addition to the system parameters, we have also used the standard quality of service classs defined in LTE and different services are supported on 9 logical channels with varying QoS [8]. In the traffic generator, we utilized the average packet size and average inter-arrival time for each service based on [9] and [10]. The actual inter-arrival time between two packets for a given service of a user (block) is affected by TLb (ranging on [0,1]) and given as:

$$\mathrm{APIT} = \begin{cases} \frac{\mathrm{PIT_b}}{\mathrm{TL_b}}, & \text{if } \mathrm{TL_b} \neq 0 \\ \text{no traffic}, & \text{otherwise} \end{cases} \tag{4}$$

For our simulation results in Fig. 3, 4 and 5, $\mathrm{TL_b}$ is generated randomly. Also the subscription type for every user is randomly chosen from basic, silver and gold.

B. Results

We compare the performance of our proposed scheduler with three traditional scheduling algorithms: round-robin, proportional fair and maximum-throughput [3]. Three major performance metrics of system throughput, fairness and satisfaction of GBR (providing promised Qos) are compared. In Fig. 3, we have compared the system throughput against

number of active users in the system. It can be seen that the performance of proposed scheduler is better than all the other algorithms. We have also plot the theoretical system throughput to set a benchmark.

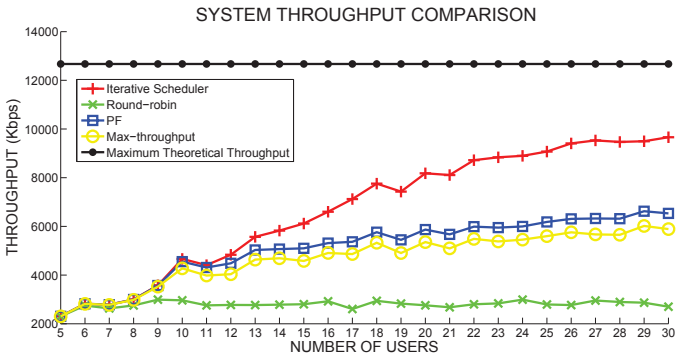

Fig. 3. Throughput Comparison

In addition, we have plotted the fairness index and the satisfied GBR percentage in Fig. 4 and 5 respectively. The satisfied GBR percentage refers to the percentage of the logical channels (services) that are served with a data rate greater than or equal to GBR. Fig. 4 and Fig. 5 show that the proposed scheduler not only increases the throughput but also results in improved fairness index and provides better quality of service to users with higher percentage of satisfied GBR. Even with large number of active users, the fairness index and percentage of satisfied users is quite significant.

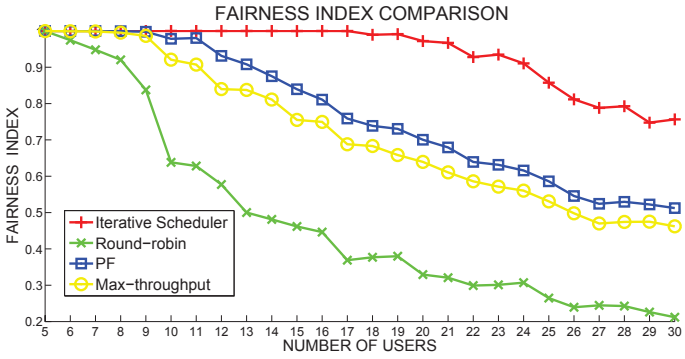

Fig. 4. Fairness Index

V. Conclusion

In this paper, we proposed a MAC-layer iterative scheduler that is characterized by sorting on the basis of three parameters and allocating resources to satisfy promised QoS to multiple real-time service of every user. In addition, it utilized the concept of converting two-dimensional buffer into a single dimension that results in higher resolution of scheduling. The comparative analysis showed that the proposed scheduler performs better that the traditional scheduling algorithm in terms of system throughput as well as fairness and satisfaction of users.

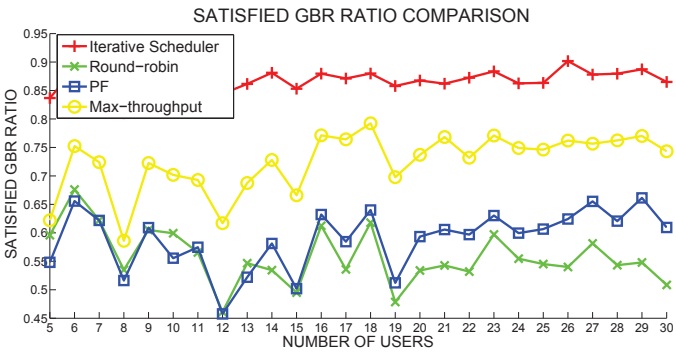

Fig. 5. Satisfied GBR Ratio

Acknowledgment

The research leading to these results has received funding from Finnish Funding Agency for Technology and Innovation (TEKES), Efore Oyj, European Communications Engineering, as well as European Research Council under the Seventh Framework Programme (FP7/2014- 2017) grant agreement 612050 for FLEX project.

References

[1] The Statistics Portal, Retreived from *http://www.statista.com/topics/779/mobile-internet/chart/1380/mobile-web-usage/*

[2] Cisco Visual Networking Index: Forecast and Methodology, 20122017, Retrieved from *http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf*

[3] Capozzi, F.; Piro, G.; Grieco, L.A.; Boggia, G.; Camarda, P., Downlink Packet Scheduling in LTE Cellular Networks: Key Design Issues and a Survey, *IEEE Communications Surveys and Tutorials*, vol.15, no.2, pp.678,700, Second Quarter 2013

[4] Zaki, Y.; Weerawardane, T.; Gorg, C.; Timm-Giel, A., Multi-QoS-Aware Fair Scheduling for LTE, *IEEE Vehicular Technology Conference (VTC Spring)*, vol., no., pp.1,5, 15-18 May 2011

[5] Chaudhuri, S.; Das, D.; Bhaskaran, R., Study of advanced-opportunistic proportionate fairness scheduler for LTE medium access control, *IEEE International Conference on Internet Multimedia Systems Architecture and Application (IMSAA)*, vol., no., pp.1,6, 12-13 Dec. 2011

[6] Bhamri, A.; Nikaein, N.; Kaltenberger, F.; Hämäläinen, J.; Knopp, R., Pre-processor for MAC-layer Scheduler to Efficiently Manage Buffer in Modern Wireless Networks, submitted to *IEEE Wireless Communications and Networking Conference (WCNC)*, April 2014

[7] 3GPP TS 36.212. Evolved Universal Terrestrial Radio Access: Multiplexing and Channel Coding. ver. 8.6.0, March 2009. URL http://www.3gpp.org/ftp/specs/html-info/36212.htm

[8] Alasti, M.; Neekzad, B.; Jie Hui; Vannithamby, R., Quality of service in WiMAX and LTE networks [Topics in Wireless Communications], *IEE Communications Magazine*, vol.48, no.5, pp.104,111, May 2010

[9] CISCO, Voice Over IP - Per Call Bandwidth Consumption, Retrieved from *http://www.cisco.com/image/gif/paws/7934/bwidth_consume.pdf*

[10] CISCO, Implementing QoS Solutions for H.323 Video Conferencing over IP, Retrieved from *http://www.cisco.com/image/gif/paws/21662/video-qos.pdf*