Norwegian University of
Science and Technology

Packet Scheduling Algorithms for Wireless Networks

Bjørn Hovland Børve

Master of Science in Communication Technology
Submission date: June 2008
Supervisor: Geir Egil Øien, IET
Co-supervisor: Mikael Gidlund, Nera Networks

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

Problem Description

The project will study scheduling in wireless high-speed networks, taking into account QoS requirements. Modern radio-links transport many different classes of data, each exhibiting different QoS requirements. Good scheduling schemes should be able to take these QoS requirements into account when handeling traffic. By differentiating between data-flows which have different QoS requirements a better utilization of the available resource can be made. The work to be done in this report is an analysis of different scheduling algorithms for high-speed radio-links.

Assignment given: 15. January 2008
Supervisor: Geir Egil Øien, IET

Abstract

The evolution of services offered using communication technology has yielded a jungle of different services. Many of these services exhibits different QoS requirements; different requirements to delay, probability of packet-loss and throughput. Effectively this means they require different amounts of resources when handled in a network node. This thesis covers descriptions and simulations of four different scheduling algorithms deployed in a high-speed point-to-point radio-link scenario. The different algorithms examined in this thesis are the conventional First-In-First-Out (FIFO) algorithm, the Strict Priority (SP) queuing algorithm, the Deficit Round-Robin (DRR) algorithm and finally the Deficit Weighted Round-Robin (DWRR) algorithm. Theoretical presentations of each of the algorithms are followed by simulations which exhibits the characteristics of the different algorithms. Two of the mentioned algorithms (FIFO, DRR) does not offer capabilities of differentiation between different classified data-flows, while the two remaining algorithms (SP, DWRR) does. The simulations illustrates the advantages of deployment of scheduling algorithms capable of differentiating resource allocation with respect to the different QoS requirements of multiple data-flows. The simulation results reveals that when deployed in a high-speed point-to-point link, where low complexity is emphasized, the DWRR algorithm offers the most promising performances of the examined algorithms.

Preface

This thesis was completed as a part of the 5-year Master's Degree Program and is the final examination for the Master's Degree in Communications Technology at Norwegian University of Science and Technology (NTNU), Trondheim, Norway. The thesis was completed in conjunction with Nera Networks, Bergen.

The project offered an opportunity to examine scheduling algorithms that could potentially be implemented in future generation networks. Personally I find this topic interesting, especially the dynamic features of algorithms with capabilities of differentiation between data-flows are appealing to me.

I would like to thank Mikael Gidlund at Nera Networks and Professor Geir E. Øien at NTNU for guidance and feedback throughout the work with this thesis. Furthermore would I like to thank Bård Henriksen at Nera Networks for helping me with the simulation environment.

Bjørn Hovland Børve
Trondheim 10. June 2008

Abbreviations

Term	
AF	Assured Forwarding
ARQ	Automatic Repeat-reQuest
BE	Best Effort
CNR	Carrier to Noise Ratio
DRR	Deficit Round Rubin
DWRR	Deficit Weighted Round Rubin
EF	Expedited Forwarding
FIFO	First-In-First-Out
GPS	Generalized Processor Sharing
MCS	Maximum Carrier to Noise Ratio Scheduling
ORR	Opportunistic Round Robin
PFS	Proportionally Fair Scheduling
RPS	Rate-Proportionally-Servers
RR	Round Robin
SE	Scheduling Entity
SP	Strict Priority Queuing
TDM	Time Division Multiplexing
ToS	Type-of-Service
WFQ	Weighted Fair Queuing
WF2Q	Worst-case Fair Weighted Fair Queuing
WF2Q+	Worst-case Fair Weighted Fair Queuing +
WFS-ARC	Weighted Fair Scheduling based on Adaptive Rate Control
WRR	Weighted Round Robin
QoS	Quality of Service

Contents

List of Figures

Chapter 1

Introduction

Algorithms used for scheduling of packet transmission in wireless communication networks is a hot topic for research. Capabilities of differentiation between different classified data-flows is of great interest in the growing jungle of services offered over both wireless and wired platforms. The following thesis describes and discusses four different scheduling algorithms.

1.1 Motivation

The motivation behind this thesis is the growing demand of effective scheduling algorithms which are capable of differentiation between different data-flows. Different services offered have different requirements with respect to delay, probability of packet-loss, throughput, etc. Algorithms that exploits these different requirements in an effective manner is thus of great interest.

1.2 Project Context

This project is a master thesis carried out in collaboration between NTNU and Nera Networks. Mikael Gidlund of Nera Networks proposed the assignment, Professor Geir E. Øien has offered guidance on behalf of NTNU, while several other employees of Nera Networks have contributed throughout constructive discussions.

Nera Networks is a major player in the telecommunications industry in Norway, employing around 800 worldwide. The headquarter of Nera Networks is in Bergen, while they have regional offices throughput the world.

1.3 Problem Description

The focus of this assignment was originally on an analysis of the behavior of scheduling algorithms deployed in fast fading environments. Throughout discussions with the founders of the assignment, who indicated it to be a very open assignment, it later evolved to focus on the features of a selection of algorithms suitable for high-speed point-to-point radio-links. Since the algorithms discussed are most suitable for high-speed point-to-point links, where fading rarely is fast, cross-layer algorithms exploiting channel state informations is omitted from the main parts of the thesis, only to be mentioned in chapter 2.6, where several other interesting algorithms are presented in a very compact manner.

This thesis will focus on the mentioned high-speed point-to-point radio-link scenario. A selection of four different algorithms with low complexity are to be described and simulated with focus on performance measures such as delay, throughput and loss-rate. The low complexity of the algorithms is a important feature of the selected algorithms since low complexity is important in high-speed network nodes. Furthermore, the capabilities of some of the different algorithms to differentiate the resource allocation between different classified data-flows will be emphasized. The different algorithms are furthermore to be tested in environments including ill behaving sources, fading, and shortage/abundance of queue capacity.

1.4 Thesis Outline

The following thesis is organized as follows: In the next chapter, Chapter 2, a further introduction to scheduling is given. The term QoS is defined together with several performance parameters, and an elaborate description of the simulation setups used throughout the thesis is presented. Furthermore, brief descriptions of other proposed algorithms together with literature references is given in the final part of Chapter 2. In chapter 3 the First-In-First-Out (FIFO) algorithm is described and its corresponding simulation results is presented and discussed. Chapter 4 presents the Strict Priority (SP) algorithm, and following the same framework as for Chapter 3, the simulation results is presented and discussed. The same approach is used when presenting the Deficit Round Robin (DRR) algorithm and the Deficit Weighted Round Robin (DWRR) algorithm in Chapter 5 and Chapter 6 respectively. In Chapter 7 the influence which different queue capacities has on the performances of the different algorithms is presented, while Chapter 8 focuses on the impact fading has on the performance of the different algorithms. The main discussion and conclusion of the thesis as whole can be found in Chapter 9.

Chapter 2

Scheduling

The recent trends in communications technology concerning convergence of different services to terminals offering a multiply of services leads to a new flora of traffic patterns compared to the former traffic patters of services designed for best-effort performances. Several different services are demanded from the same terminal, where each of these different services exhibit individual demands of resources. The multiple services transported to the terminals often share the same link at some point in the networks. This leads to a growing demand of networks capable of differentiating between different traffic-classes, in other words network nodes offering Quality-of-Service (QoS) support. In the following sections different terms for QoS are defined, before a description of the scenario used when simulating the different scheduling algorithms is made. Furthermore, parameters which are of interest when comparing the different algorithms to one another are defined. The final section gives an introduction to the simulation tools used throughout the project.

2.1 Quality-of-Service

Different services require different constraints with respect to several parameters. Four primary parameters can be defined: reliability, delay, jitter and bandwidth. Examples of services requiring different QoS might be e-mail, having high reliability requirements, but rather low requirements with respect to delay, jitter and bandwidth. On the other hand, videoconferencing would have low reliability demands, a lost packet in a video-stream is not a catastrophe, while requirements to delay, jitter and bandwidth would be high [3]. If an entity offers dynamical resource allocation with respect to the different requirements from a non-homogeneous selection of entities, the entity can be defined as an entity offering QoS support [2].
The resources made available for allocation might be power, bandwidth, transmission time, etc. The focus on allocation in this theses will be with respect to transmission time, as the differentiation between the different entities will be made in scheduling of packets from a queue transmitted over a single link. Furthermore, the traffic is divided into three different Type-of-Service (ToS) classes, dependent of given QoS requirements. The classes used are:

- Best-Effort (BE) - A node offering BE service simply uses resources when made available,

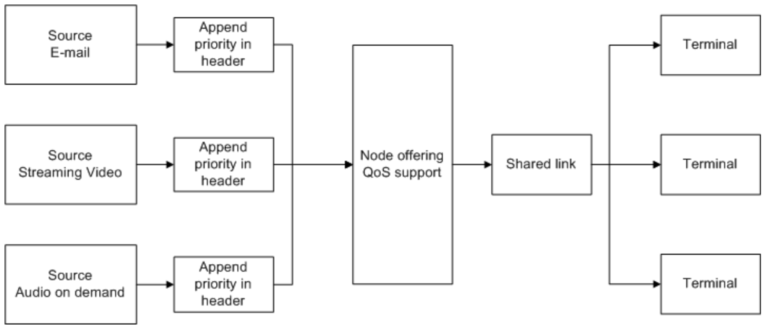


Figure 2.1: **Example of scenario where different data-flows are classified, before a QoS-capable network node allocates resources dynamical dependent on the different prioritized data-flows.**

without offering any guarantees with respect to data delivery or to fulfillment of any QoS-requirements. BE is the only service used in older conventional network nodes.

- Assured-Forwarding (AF) - In an AF implementation four different priority classes are defined within the AF-domain. Each of the classes is allocated a part of the resources which are available. Within each of the four classes another prioritizing is defined, three different parameters dependent on the probability for that a packet would be discarded undergoing congestion. All together this potentially makes it twelve different service classes [3].
- Expedite-Forwarding (EF) - Consider a scenario where two types of data-flows are using the same link. If one of the data-flows is classified as expedite and the other one as regular, the expedited data-flow should be able to transit the network as if packets classifies differently where not present at all. For an example, the total bandwidth available could be divided into two parts, where the data-flow classified expedite would be reserved an amount of bandwidth sufficient to handle the incoming expedite data-flow without congestion/delay [3].

In the next section the simulation scenario used for testing algorithms when using the terms mentioned above as building-blocks in a scheduling scenario. Note that the properties of the definitions made above are not emphasized in this thesis, the main feature of the different classifications used in the simulations are simply the classifications them self. To make it possible to differentiate between flows a form of tagging is needed, the tagging used for differentiation is the mentioned classifications.

2.2 Simulation Scenario

Building on the scenario described in Figure 2.1, a bit altered scenario is defined in Figure 2.2. The different sources feeds data onto a scheduling entity (SE) which first routes the the data-flows dependent of the different ToS, either BE, AF or EF. In the simulations the three different sources are initially made identical, transmitting with identical average rate and packet size. The rate of the sources are Poisson distributed, with equal average and individual starting seed.

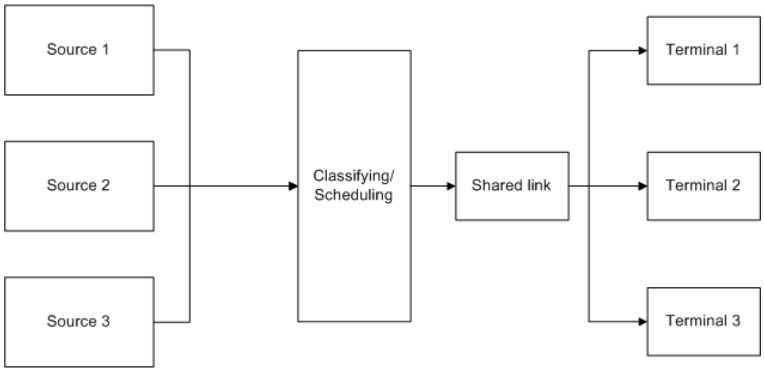


Figure 2.2: **Overview over the different entities used in the simulations. The sources transmit data onto the scheduling entity which routes the different flows dependent of their ToS , before it schedules and transmits the data over the shard link.**

The Poisson distribution can be defined as follows: If the expected number of occurrences in a time interval is λ, then the probability that there are exactly k occurrences is equal to

$$f(k,\lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{2.2.1}$$

where k is the number of occurrences of an event, e is the base of the natural logarithm and λ is a positive real number equal to the expected number of occurrences that occur during the given interval.

The packet size parameter is an important parameter since a scheduling algorithm could potentially use packet counting as a parameter in resource allocation. If the packet sizes where different from a data-flow to another, a data-flow containing larger packets would potentially have an advantage since more bits would be transmitted for each packet. This issue can be resolved by instead using byte or bit counting in the scheduling algorithm.

The classifying of the data-flows could be done at the source (Fig.2.1), locally within the scheduler (Fig.2.2, Fig.2.3) or in a combination of the two. A given company offering a given service would of course want their service to be prioritized in such a manner that the experience the user of the service gets is as least as good promised. But on the other hand, if the given service traverses a network owned by a company that wants to prioritize the data-flows in a different manner, another classifying of the data-flows could be applied. This issue is more of a network policy matter, and will not be discussed any further in this theses. For simplicity, the data-flows are classified with a given ToS at each source only, and handled in the SE dependent of the ToS classification of a given data-flow. As seen in Figure 2.3, the three identical data-flows arrive at the SE where the different data-flows are routed to their respective queues dependent of their ToS classification. Note that the AF-classified data-flow is passed on to a sub-classifier which classifies the arriving data-flow within its own sub-domain. As mentioned in Section 2.1, the AF-scheme effectively offers twelve different classifications within its sub-domain, but since only one of the data-flows is classified as an AF-flow in the SE, no intra-scheduling within the

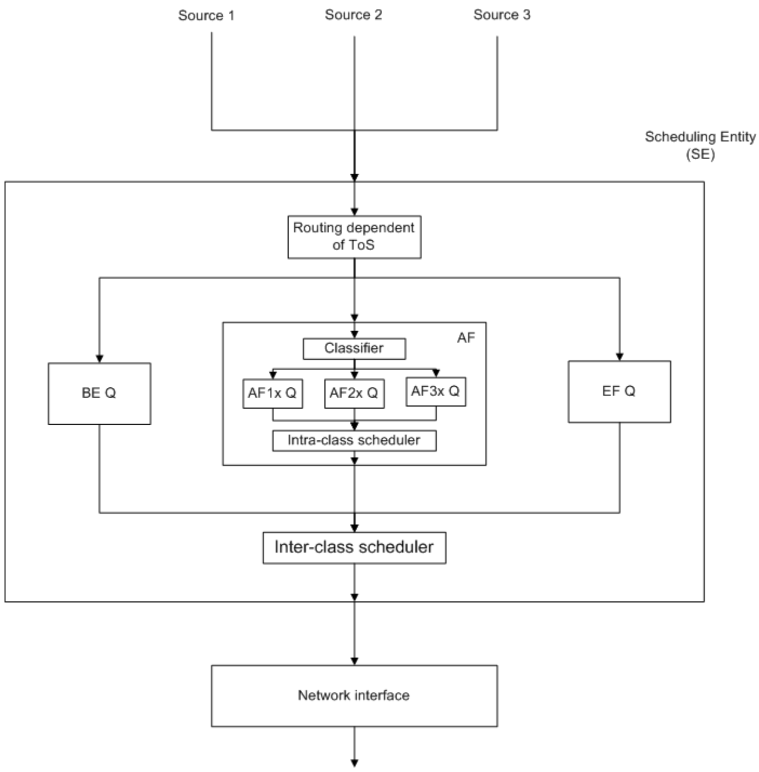


Figure 2.3: **The scheduling entity applies a scheduling algorithm upon the set of classified data-flows.**

AF-domain is needed.

The different queues in the SE are set with a given capacity and first-in-first-out (FIFO) drop-tail characteristics. In a FIFO drop-tail queue the first arrived packet is served first, all latter packets are placed in a sequential order dependent of arrival time, before being moved one step closer to the output for each packet sent. If the packets arrives faster than they are sent, an overflow could occur if the capacity of the queue is final. When experiencing such a scenario, the queue drops the latest arrived packets, thereby the expression drop-tail. Packets which are dropped are lost as far as the SE concerns. Higher-layer logics could though implement a re-transmission scheme of some sort, but further discussion of this topic is beyond the scope of this theses. A more elaborate description of FIFO is given in Chapter 3.

As seen at the bottom of Figure 2.3, a scheduling entity gathers packets from each of the classified queues, before transmitting the packets to the network interface. The scheduler could implement various scheduling algorithms, algorithms which are to be discussed in the following chapters of this theses. Note that for simplicity, the scheduler has always packets available for transmission from each of the queues in the simulations made in the following chapters, it will never experience a queue underflow.

The receiving node discriminates between the different data-flows dependent of the ToS classification and calculates the different statistics used in the further analysis.

2.3 Simulation Time-line

The simulation scenario mentioned above manipulated through alterations in key parameters throughout a simulation run. Altering the available resources and traffic patterns during the simulations makes the key characteristics of the different scheduling algorithms more distinguishable. The alterations can be divided into two parts, one where alterations in the traffic pattern of the different sources are made, and another part where alterations with respect to the available resources are made.

- Traffic pattern alteration:
  - BE ToS classified source changes its data-rate and packet-size
  - AF ToS classified source changes its data-rate and packet-size
  - EF ToS classified source changes its data-rate and packet-size
- Resource alteration:
  - SE changes the available bandwidth of the shared link

In Figure 2.4 the timing of the different alterations is described. The traffic pattern alterations are done in two different intervals, the first one from 5 to 16 seconds and the second interval from 32 to 43 seconds. From the initial setup with equal data-rate for all of the sources, the three sources increases their respective data-rates one by one.

In the interval from 5 to 8 seconds the BE ToS classified source increases its data-rate while the two remaining sources does not change their rates. After 8 seconds of the simulation time-line the data-rate of the BE source is decreased to its initial value, while after 9 seconds another source, this time the AF ToS classified one, increases its data-rate. 3 seconds later, at 12 seconds, the AF data-rate is decreased to its initial value. The same process is repeated at 13 to 16 seconds, the only difference at this interval is that it is the EF ToS source which exhibit a boost in data-rate.

In the interval from 16 to 20 seconds the parameters are back at their initial values, before a decrease in bandwidth at the SE node is applied from 20 to 25 seconds, only for a even tighter bandwidth to be applied from 25 to 28 seconds. In the following 4 seconds, from 28 to 32 seconds, the parameters are again back at their initial value.

In the interval from 32 to 35 seconds the BE ToS source reduces the size of the packets which are transmitted, before increasing the size again, to its original value. The same process is applied for the AF ToS source in the interval from 36 to 39 seconds, and for the EF ToS source in the interval form 40 to 43 seconds. In the final 2 seconds all system parameters are restored to their initial values.

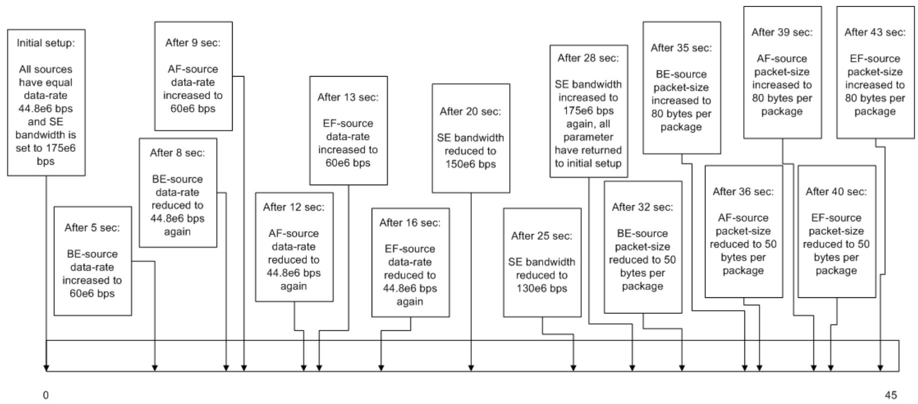


Figure 2.4: **Overview of the different setup alterations done throughout a simulation run.**

2.4 Quality Parameters

As mentioned in Section 2.1, a variety of parameters can be used when characterizing QoS. In this section the parameters which are focused upon when considering the different scheduling algorithms in the following chapters are defined.

- Delay - the difference in time between transmission from source and arrival at the destination. Various statistics such as maximum delay, minimum delay and the variance of the delay are used when the describing the delay characteristics a given scheduling algorithm exhibits. The average delay of the i'th flow can be expressed by:

$$D_i = \frac{1}{N}\sum_{j=0}^{N}\left[rt_i(j) - tt_i(j)\right] \tag{2.4.1}$$

  where $tt_i(j)$ is the time when the j'th packet of the i'th flow was transmitted from its source, $rt_i(j)$ is the time when the j'th packet of the i'th flow was received at the receiver and N is the number of packets used for calculating the average delay.

- Packet-count - the number of packets received at a given destination per second. Also counting lost packets and the loss-rate is interesting when comparing reliability.

- System throughput - the total number of bytes per second at the output of the SE. The different scheduling algorithms utilizes the total system capacity differently, the system throughput is therefore an interesting parameter when considering the utilization of the resources when deploying the different algorithms. The system throughput R can be expressed by:

$$R = \frac{1}{T}\sum_{i=0}^{N}\sum_{j=0}^{K} p_i(j) \tag{2.4.2}$$

  where T is the size in seconds of the time-window where the throughput is calculated, $p_i(j)$ is the size in bytes of the j'th packet in the i'th flow, K is the number of packets departing in the time-window T and N is the number of flows handled by the scheduler. Examining the throughput of a single flow simply yields:

$$R_i = \frac{1}{T}\sum_{j=0}^{K} p_i(j). \tag{2.4.3}$$

- Fairness - a measure for how equally the available resources are allocated among the different data-flows. A formal definition of fairness can be expressed by the Jain's fairness index, expressed by:

  $$J = \frac{(\sum_{i=1}^{N} x_i)^2}{N\sum_{i=1}^{N} x_i^2} \tag{2.4.4}$$

  where x_i is the resources allocated to user i and N is the number of participants in the system. The worst-case scenario with total unfairness yields $J = \frac{1}{N}$ and the best-case scenario with total fairness yields $J = 1$ [31].

The different parameters mentioned above are useful when describing the performances of the different algorithms when deployed upon a given scenario. In the following chapters a collection algorithms are described and compared to each others with the help of the mentioned parameters.

2.5 Simulation Environment

The scripts used for simulations are written for J-Sim, a component-based, compositional simulation environment written in Java. In the public release of J-Sim multiple frameworks are available, the simulations in this theses are mainly written within the INET Framework and the DiffServ Framework. More information concerning the J-Sim simulation environment can be found at ”http://www.j-sim.org”.

2.6 Other Algorithms

In addition to the algorithms examined in this thesis there is a flora of other scheduling algorithms proposed in various literatures. Many algorithms have been proposed, each with pros and cons in form of complexity, fairness, delay, etc. Some are specialized for transmission over orthogonal fading channel, others more effective in shard link environments. Some are cross-layer designed, using channel state information, while others are kept strictly at single layers. In the following sections some of the proposed algorithms are presented.

2.6.1 Proportionally Fair Scheduling

The Proportionally Fair Scheduling (PFS) algorithm was proposed after studying the unfairness exhibited when increasing the capacity of CDMA by means of differentiating between different users. Transmission of pilot symbols to the different users yields channel state information, and by allocating most resources to the users having the best channels, the total system capacity of the CDMA scheme could be increased. Such allocation of resources favors the users closest to the transmitting node, resulting in reduced fairness between the different users. The PFS algorithm seeks to increase the fairness among the users at the same time as keeping some of the high system throughput characteristics.

PFS uses throughput monitoring, the time a given user is allowed to transmit is logged and furthermore this log is used when prioritizing between the different users. If a given user does not transmit for a longer period of time than the other users, the priority of the non-transmitting user is increased. Using the TDM scheme the PFS algorithm can be expressed as follows: $P(t) =$ available power at time t for use by K users. $T_k(t) =$ throughput of user k over a time window up to time t, $k = 1, ..., K$. Furthermore, to find the user that has the highest priority in a time slot the following expression must be calculated for all k users:

$$J_k = \frac{(C/I)_k(t)}{T_k(t)} \tag{2.6.1}$$

where $(C/I)_k$ is the carrier to interference ratio of user k. The user that has the highest J is allowed to transmit with power $P(t)$ [5]. A mutation of the PFS algorithms is described in [9], where a weighting factor is introduced in the priority calculations expressed by 2.6.1. The altered version can be expressed by the following equation:

$$J_k = W_k(t) \times \frac{(C/I)_k(t)}{T_k(t)} \tag{2.6.2}$$

where $W_k(t)$ is the weight of the kth user. The weighting factor could either be static, where an user gets a constant weight, or dynamic, where the weights could be updated stepwise to meet given QoS and priority requirements. The weighting would thus give an increased degree of freedom when scheduling traffic. As the scope of this thesis evolved to mainly focus on high-speed point-to-point scenarios, where all users share an equal channel, a further discussion concerning the pilot symbol dependent PFS algorithm has been omitted from this thesis. Further descriptions of the PFS algorithm can be in [5], [6], [7], [8] and [9], while a variant which offers delay constraints is described in [18].

2.6.2 Weighted Fair Queuing

The Weighted Fair Queuing (WFQ) approach to packet scheduling is a very popular approach in wirebased broadband networks, offering bounded delay, guaranteed throughput and fairness among the users. An adaptation to the wireless domain is though not trivial, since the channel characteristics of a wireless channel is quite different from a wirebased channel. The conventional WFQ algorithm can be expressed as follows: Each and every packet gets a corresponding start and finishing time tag. The start time tag is calculated for the kth packet in session i by

$$S_i^k = max\left[F_i^{k-1}, V(a_i^k)\right] \tag{2.6.3}$$

where F_i^{k-1} is the finishing time tag of the previous packet in session i and $V(a_i^k)$ is the virtual time in the Generalized Processor Sharing (GPS) [48] scheduler when the kth packet from session i arrives. After the calculation of the start time tag has finished, the finishing time tag can be calculated. The calculation of the finishing time tag of the kth packet in session i can be expressed by

$$F_i^k = S_i^k + \frac{L_i^k}{\phi_i} \tag{2.6.4}$$

where L_i^k is the length of the kth packet in session i, and ϕ_i is the weight of session i. After calculating the different finishing time tags of the different sessions, the session which has the lowest finishing time chosen for transmission [11].

Different mutations of WFQ which tries to compensate for the characteristics of wireless transmission have been proposed. In [1] the essence of what WFQ tries to compensate for has been expressed as follows: If a backlogged flow f transmits with error or large delay in the interval $[t_1, t_2]$, the flow f should be compensated over a future interval $[t_1^{'}, t_2^{'}]$ when f experience better transmission conditions. Compensation for f means that f must be allocated additional channel access during $[t_1^{'}, t_2^{'}]$. By cross-layer design the use of higher layer Automatic Repeat-reQuest (ARQ) makes it possible to get information concerning channel conditions without transmission of dedicated pilot symbols. If the channel conditions are poor, many ARQs are received, and thus it should be compensated for.

The WFQ approach to packet scheduling offers many advantages over other less complex algorithms both in fairness and system capacity utilization, but since the time tag calculations require a time complexity that grows linearly with the number of sessions serviced, it is not discussed further in this thesis where the focus is mainly on high-speed links were complexity should be reduced to a minimum. Promising work on a WFQ variant based upon Rate-Proportionally-Servers (RPS) that exhibits lower complexity has been proposed in [12], but a conversion to the wireless domain with its required channel state compensations is yet to be defined. Further literature concerning WFQ can be found in [11], [12], [13], [14], [16] and [17]. Another variant with reduced complexity called Huffmann Fair Queuing (HuFQ) is described in [15], while an algorithm offering an improved delay guarantee at the cost of a slight reduction of bandwidth fairness is proposed in [19]. Versions of WFQ which are proposed for the wireless domain can be found in [1], [20], [21] [22] and [27].

2.6.3 Worst-case Fair Weighted Fair Queuing

WFQ described in the previous section is an emulation of the idealized GPS. A GPS does not transmit whole packets, it assumes that all sessions can be served simultaneously and that all entities are infinitely dividable. In practice only whole packets can be transmitted and only one session can be served at the time. WFQ is a unideal practical realization of the GPS, its worst-case fairness has been shown to be much weaker than that of GPS [1]. Worst-case Fair Weighted Fair Queuing (WF2Q) uses another packet approximation of GPS that shares both the bounded-delay and the worst-case fairness of GPS. Instead of the calculation the finishing time tag for all packets at a given time τ as done in WFQ, the WF2Q approach does only calculates the time tags of the packets which would either be served currently or already have been served in the corresponding GPS system at the given time τ. The packet among them that has the shortest finishing time would than be serviced first. This means that the WF2Q approach is a more idealized approximation to GPS than WFQ. The complexity WF2Q is though high when compared to the low complexity alogorithms discussed in detail in this thesis. Further information concerning WF2Q can be found in [23] and [16], while a delay optimized WF2Q variant is described in [24].

Another variant of the WFQ approach called Worst-case Fair Weighted Fair Queuing + (WF2Q+) is described in [16]. WF2Q+ provides the same delay bound and fairness as WF2Q, but with a lower complexity than WF2Q. The main difference from WFQ and WF2Q to WF2Q+ is the use of a new system virtual time function that achieves both low complexity (O(log(n)) and high accuracy in approximating the ideal virtual time function of GPS. WF2Q+ is described in [16].

2.6.4 Round-Robin Algorithms

The algorithms described in chapter 5 and 6 are only two of several Round Robin (RR) based algorithms. Other algorithms such as the Opportunistic Round-Robin (ORR) scheduling algorithm have been proposed. ORR is capable of exploiting multiuser diversity at the same time as short-term fairness is remained [10]. Another approach is the List-Based Weighted Round Robin (WRR) where the sessions are served as defined in a list. The list contains session identities, and the number of times the identity of a given session appears in the list is proportional to the weight of the given session [25].

2.6.5 Other Algorithms

It is also possible to deploy different algorithms together in a hierarchical structure. In [28] a structure deploying both Strict Priority (SP) and WFQ in a single scheduling entity is proposed.

In [29] a cross-layer approach named Weighted Fair Scheduling based on Adaptive Rate Control (WFS-ARC) is described. WFS-ARC dynamically adjusts the data rate parameters of the MAC layer based upon channel state information gathered by the PHY layer. Above the MAC layer, the LLC layer schedules the packet transmission opportunistically to the one user that has best channel conditions and that also satisfy fairness constrains.

When deployed in a Time Division Multiplexing (TDM) scenario, the Maximum Carrier to Noise Ratio Scheduling (MCS) algorithms simply chooses the user that has the highest Carrier to Noise Ratio (CNR) in each time-slot, and the number of time-slots allocated to a user within K time-slots is distributed by a binomial distribution [30].

2.6.6 Comments

Of the different mentioned algorithms, it is the Round-Robin based algorithms which are of most interest when considering high-speed networks. The low complexity of the Round-Robin family (O(1)) makes them efficient and simple to implement when compared to the WFQ based algorithms which all have complexity of either O(n) or O(log(n)). They do though offer greater fairness and better bandwidth utilization, and are of great interest when considering other scenarios than the high-speed point-to-point scenario. The cross-layer design of WFS-ARC and the pilot based PFS are also interesting algorithms when considering scenarios where multiple users transmits over orthogonal channels.

Chapter 3

First-In-First-Out

First-In-First-Out (FIFO) scheduling is the least complex algorithm used for scheduling examined in this thesis. The straight forward queue handling leaves little room for dynamical resource allocation, but because of the very low complexity it is the most common queue management algorithm used in conventional network nodes.

3.1 Why FIFO?

There are many well-known algorithms which have dynamical resource allocation and offers QoS support. FIFO in its primal form does not offer either dynamical resource allocation or QoS support. So why should FIFO be mentioned in a thesis considering dynamical resource allocation algorithms? Since FIFO is a very common approach to queue management, commonly used in conventional best-effort network nodes, it interesting to use results from FIFO as a form of benchmark to compare the results of the other algorithms with.

3.2 Algorithm Description

The FIFO algorithm is very simple. As seen in Figure 3.1, the principle is that the packet arriving at the queuing entity first, will also be the first packet to leave the queue. Akin to for example a queue of costumers being served by a single teller in a bank, the costumers who arrived first will be served first and so on. One could perhaps argue that the clerks should prioritize costumers with major contributions to the income of the bank first, and rightly so, this is a major weakness of the FIFO queue approach. If packets were to arrive at a higher rate than the output rate of the scheduler over some period of time, the capacity of the queue could be exhausted. This scenario could be handled in differently; the queuing entity could tell one or several of the data-sources to slow down for a while, or, in simple best-effort manner, just drop the packets arriving to an overflowed queue. The latter approach is the one used in the simulations throughout this thesis when considering FIFO overflow.

An illustrative implementation of the the simple FIFO approach in pseudo-code:

```
create FIFOQueue Q
```

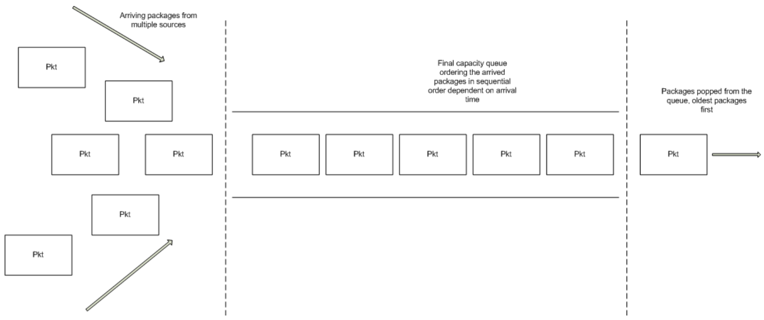


Figure 3.1: **In FIFO queuing the arriving packets are placed into a sequential order, dependent of the arrival time of each packet. The packet which arrived first will be dequeued first, in other words, the oldest packet will always be sent first.**

```
void enqueue(Packet pkt)
      Q.append(pkt)

void dequeue()
    if(Q.isempty() == false)
      temppkt = Q.removepkt()
    send(temppkt)
```

3.3 Simulations

In this scenario the three data-flows are fed onto a shared FIFO queue which does not differentiate between the data-flows at all. As mentioned in Section 3.1, the simulation results from the FIFO scheduled scenario offers little surprises. The lack of differentiation between the different ToS classified data-flows makes the simulations results, intuitively, more or less identical from one data-flow to another. The value of the FIFO-based simulations is merely that they can be used as benchmarks for which that simulation results from simulations where other scheduling algorithms are applied can be compared.

Some of the main parameters of the network nodes were set constant throughout the simulation, while others were altered to simulate different traffic patters and channel capacities. The main parameters are presented in Table 3.1, where only points along the time-line where alteration of parameters are made is listed. The interval between two mentioned time-line points are to be considered having static parameters within the given interval. Some of main parameters, such as the bandwidth of the links between the different sources and the SE, which are not manipulated throughout the simulation are either not mentioned at all or only marked by ”-” in Table 3.1.

Time [sec]	Node	Data-rate [Mbps]	Bandwidth [Mbps]	Packet-size [bytes]
0	BE-source	44.8	-	80
0	AF-source	44.8	-	80
0	EF-source	44.8	-	80
0	SE	-	175	-
5	BE-source	**60**	-	80
8	BE-source	**44.8**	-	80
9	AF-source	**60**	-	80
12	AF-source	**44.8**	-	80
13	EF-source	**60**	-	80
16	EF-source	**44.8**	-	80
20	SE	-	**150**	-
25	SE	-	**130**	-
28	SE	-	**175**	-
32	BE-source	44.8	-	**50**
35	BE-source	44.8	-	**80**
36	AF-source	44.8	-	**50**
39	AF-source	44.8	-	**80**
40	EF-source	44.8	-	**50**
43	EF-source	44.8	-	**80**

Table 3.1: Parameter alterations during the simulation

The sources have all Poisson distributed data-rates with individual starting seed, yielding some variation between the different sources, though with identical average rate. For more information concerning the simulation time-line, see section 2.3.

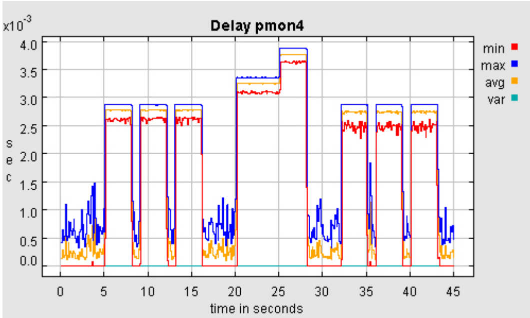
The capacity of the FIFO drop-tail queue within the SE is 60 kB. The total memory used for queuing when simulating this algorithm is set to be identical to the total memory used for queuing when simulating the other algorithms in this thesis.

The three figures 3.2a, 3.2b, 3.3a shows more or less identical delay characteristics for the three different classified data-flows. No major differences can seen, which is not a surprise, as the three data-flows have the same amount of resources available at any given time.

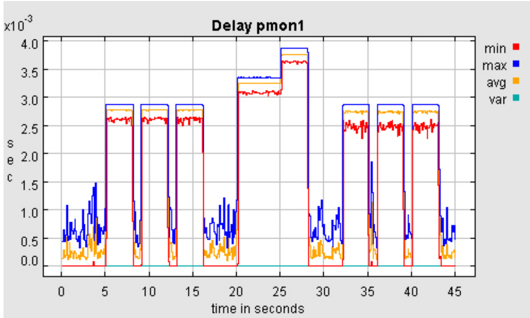
The increases in delay in the interval from 5 to 16 seconds can be explained by the increase in data-rate from each of the respective sources, leaving less resources available per packet.

In the interval from 20 to 28 seconds the increase in delay is due to the reduction of available bandwidth at the output of the SE, which then intuitively yields more congestion.

The introduction of smaller pakets in the interval from 32 to 43 seconds leaves more congestion which also introduces another increase in delay. In Figure 3.3b the maximum delay value of each

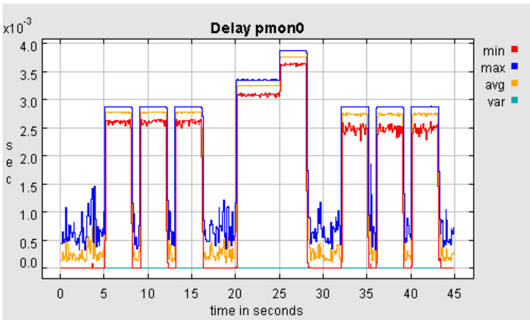


(a) BE-flow

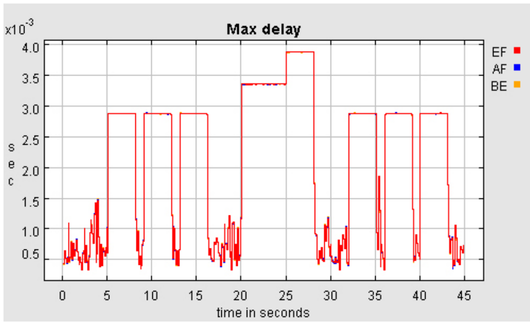


(b) AF-flow

Figure 3.2: **Delay statistics of the BE classified data-flow and of the AF classified data-flow.**



(a) EF-flow



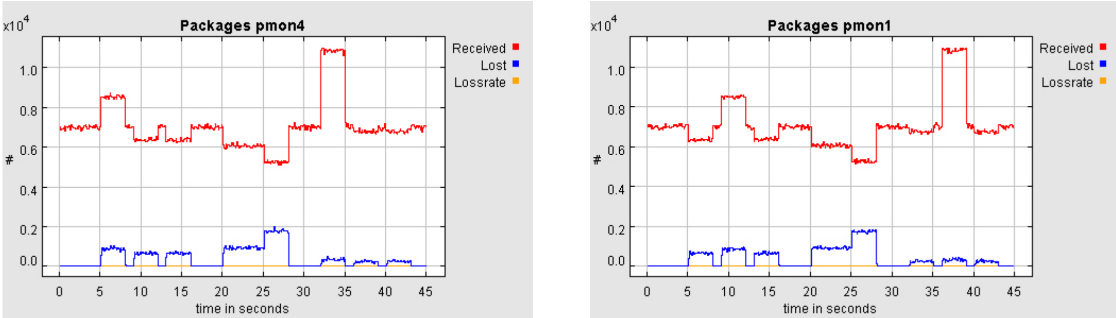
(b) Maximum delay

Figure 3.3: **Delay statistics of the EF classified flow and the maximum delays from each of the different classified data-flows.**

of the different data-flows at given time are collected in one plot. As seen in figures 3.2a, 3.2b and 3.3a, there is minimal variation between the different classified data-flows.

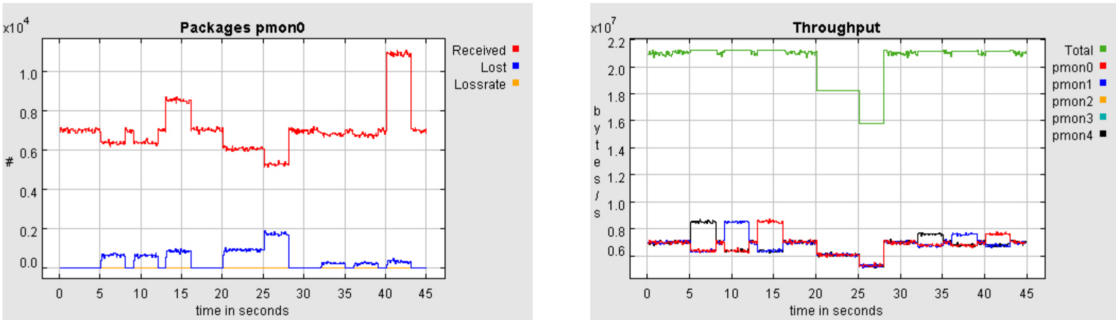
Figures 3.4a, 3.4b and 3.5a shows the packet-count of each of the different data-flows. Similarly to, and of the same reasons of the delay statistics, there are little difference from one data-flow to another.

When one source increases its output data-rate and thereby increase its received packet rate, though not equally much, the two other data-flows experience a corresponding reduction in their received packet rate. This effect can be seen in the interval from 5 to 16 seconds. When the output bandwidth of the SE is reduced in the interval from 20 to 28 seconds, all tree data-flows experience a identical decrease in resources available, which again yields identical reduction in received packet rate.



(a) BE-flow (b) AF-flow

Figure 3.4: **Received packets at the receiver, loss-rate and packets lost during transmission for the BE classified flow and the AF classified flow.**



(a) EF-flow (b) Throughput

Figure 3.5: **Received packets at the receiver, loss-rate and packets lost during transmission for the EF classified flow. At the right the total system throughput is plotted.**

In the interval from 32 to 43 seconds the effect of reducing packet-size transmitted from each of the sources is describes. Reducing the output packet-size of one specific source means that more packets from this specific source will arrive at the FIFO queue when compared to the others. A higher packet rate yields a higher probability for a corresponding packet to be at the front of the FIFO queue, which again means that the received packet rate will increase. The increase in packet rate for one source will reduce the packet-rate of the two other sources, but as seen in the plots, the reduction is not severe when applying the specific values mentioned here. For all of the different alterations there is a increase in loss-rate, this is mainly due to lack of SE bandwidth capacity, further discussed below.

In Figure 3.5b both the throughput of each of the individual sources and the total throughput are plotted.

The increase in data-rate at the output of the different sources applied in the interval 5 to

16 seconds yields a corresponding increase in the data-rates at the receivers of the individual data-flows, at the cost of a decrease in the received data-rate for the data-flows which at a given time only exhibits the initial data-rate. As seen in the plot, the maximum throughput of the SE, set by the SE output bandwidth parameter, is reached when applying the increased data-rates. Since the sum of the data-rates of the three data-flows is greater than the maximum capacity when applying the data-rate boosts, an increased loss-rate is experienced for all of the sources in the mentioned interval.

When the SE network interface bandwidth is reduced in the interval from 20 to 28 seconds, the throughput is identically reduced for each of the data-flows since they all share the same amount of resources equally.

The reduction of packet sizes applied from 32 to 43 seconds yields an increase in throughput for the given source which at a given time transmits with reduced packet size. The increased throughput comes at the cost of, as which were the case for the increased data-rate scenario mentioned above, the other sources which does not transmit with reduced packet size. Again the plot indicates that the total capacity is exhausted, leaving increased congestion and an increase in the packet loss-rate.

3.4 Further Readings

The FIFO algorithm is a well known approach to scheduling. In the following section a collection of literature where FIFO can be studied further is given. In [32] the worst case end-to-end response time and jitter when applying FIFO scheduling to a expedited forward classified flow is examined, while [33] discusses the support of FIFO to expedited forwarding per-hop behavior together with more complex time-stamp based algorithms. The low fairness, poor utilization and other problems with FIFO are studied in [34]. A further description of the basics of FIFO is given in [35].

3.5 Preliminary Conclusions

The FIFO approach is, as mentioned, merely mentioned for comparison and benchmarking. It offers no differentiation between flows and has no guarding features against ill behaving sources. Increasing the data-rate of one flow, and all the other flows experience reduced performances, decrease the packet-sizes of one flow, and all the other flows experience reduced performances. It only positive feature is its extremely low complexity.

Chapter 4

Strict Priority Queuing

In Strict Priority queuing (SP) the different data-flows are prioritized strict, where the highest prioritized queue is served first, then the next-lower priority queue and so on. The SP approach has low complexity and offers partial QoS support.

4.1 Algorithm description

The arriving data-flows are routed to FIFO queues internally in the SE, corresponding to their respective ToS classification (Fig. 2.3). At the output of the SE, the inter-class scheduler always seeks the highest prioritized queue where a packet is present and ready for transmission. A significant drawback to the conventional SP approach is that, if not modified against, the SP implemented scheduler could, in a given scenario where highest prioritized packets arrive at rate higher than the output bandwidth of the scheduler, give all available bandwidth to the highest prioritized data-flow, leaving no spare resources for lower prioritized data-flows. This results in a total breakdown in traffic for these data-flows. Queue overflow is handled by the drop-tail approach, dropping packets arriving to a queue with exhausted capacity.

An illustrative implementation of the SP approach in pseudo-code:

```
create FIFOQueue Qpri1, Qpri2, Qpri3

int priority()
    return priority

void enqueue(packet pkt)
    if(pkt.priority == 1)
      Qpri1.append(pkt)
    if(pkt.priority == 2)
      Qpri2.append(pkt)
    if(pkt.priority == 3)
      Qpri3.append(pkt)

void dequeue()
```

```
if(Qpri1.isempty() == false)
  temppkt = Qpri1.removepkt()
elseif(Qpri2.isempty() == false)
  temppkt = Qpri2.removepkt()
elseif(Qpri3.isempty() == false)
  temppkt = Qpri3.removepkt()
send(temppkt)
```

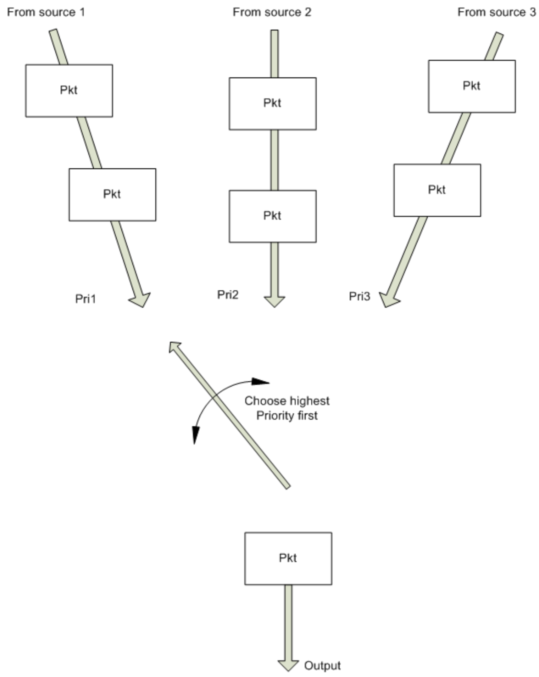


Figure 4.1: **The scheduling entity routes the different data-flows dependent of their respective ToS. At the output packets from the highest prioritized data-flow are chosen first, then next-level prioritized packets and so on.**

Another way to describe the SP algorithm is shown in Figure 4.1. The three different data-flows are fed onto their respective ToS dependent queues, before the scheduling entity, shown as a simple switch in the figure, chooses the one packet with highest priority that is available for transmission.

The SP algorithm does as described, offer differentiation between different classified data-flows, an aspect central when trying to achieve the respective QoS requirements of the different flows. Although differentiation is supported, SP can not be reckoned as an algorithm offering full QoS support. The conventional SP approach is not very dynamical and has few degrees of freedom. For an example, services which might require a guaranteed lower-bound bandwidth without having an overall data-rate requirement justifying first prioritizing does not get a high degree of supported from a SP implemented node.

4.2 Simulations

In the simulation scenario for the SP-based SE, the SE routes the three data-flows onto three different FIFO drop-tail sub-queues, dependent of the ToS classification of the respective flows. The mentioned sub-queues are identical, with a capacity of 20000 bytes. Note that the differentiation parameter, the queue priority tag, is distributed from one to three, where the data-flow which gets the tag three is the highest prioritized data-flow and the one which gets the tag one is the lowest prioritized data-flow.

- BE-queue weight: 3
- AF-queue weight: 2
- EF-queue weight: 1

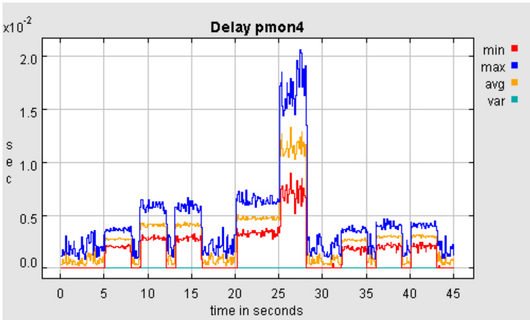
The main parameters are presented in Table 4.1, where only points along the time-line where alteration of parameters are made is listed. The interval between two mentioned time-line points are to be considered having static parameters within the given interval. Some of main parameters, such as the bandwidth of the links between the different sources and the SE, which are not manipulated throughout the simulation, are either not mentioned at all or only marked by "-" in the Table 4.1.

Time [sec]	Node	Data-rate [Mbps]	Bandwidth [Mbps]	Packet-size [bytes]
0	BE-source	44.8	-	80
0	AF-source	44.8	-	80
0	EF-source	44.8	-	80
0	SE	-	175	-
5	BE-source	**60**	-	80
8	BE-source	**44.8**	-	80
9	AF-source	**60**	-	80
12	AF-source	**44.8**	-	80
13	EF-source	**60**	-	80
16	EF-source	**44.8**	-	80
20	SE	-	**150**	-
25	SE	-	**130**	-
28	SE	-	**175**	-
32	BE-source	44.8	-	**50**
35	BE-source	44.8	-	**80**
36	AF-source	44.8	-	**50**
39	AF-source	44.8	-	**80**
40	EF-source	44.8	-	**50**
43	EF-source	44.8	-	**80**

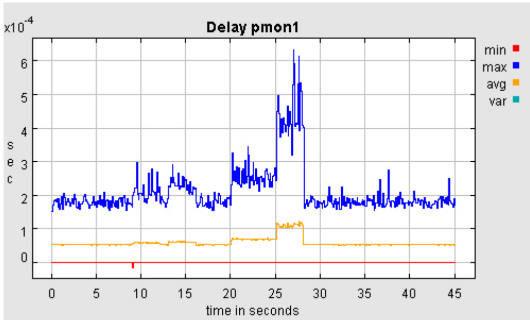
Table 4.1: Parameter alterations during the simulation

The sources have all Poisson distributed data-rates with individual starting seed, yielding some variation between the different sources, though with identical average rate. For more information concerning the simulation time-line, see section 2.3. The capacity of each of the FIFO drop-tail queues within the SE is 20 kB, adding up to a total of 60 kB for all three queues. The total memory used for queuing when simulating this algorithm is thus identical to the total memory used when simulating the other algorithms in this thesis.

4.2.1 Delay Consideration



(a) BE-flow



(b) AF-flow

Figure 4.2: **Delay statistics of the BE classified data-flow and of the AF classified data-flow.**

In figures 4.2a, 4.2b and 4.3a the delay statistics of the different ToS classified data-flows are plotted. As seen in the different plots, there is a great difference in delay from one ToS classified flow to another. As mentioned in Section 2.3, the sources exhibits, one by one, an increase in data-rates in the interval from 5 to 16 seconds.

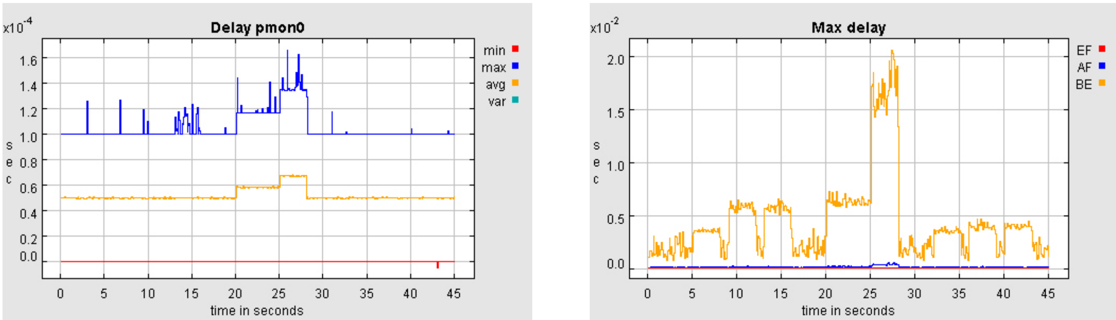
The first source which increases its data-rate is the BE classified flow. This occurs in the interval from 5 to 8 seconds. Figure 4.2a shows a slight increase in overall delay for the BE classified flow itself in this interval, while the other data-flows (Fig 4.2b, 4.3a) experiences no increase in delay at all.

In the interval from 9 to 12 seconds the AF ToS classified flow increases its data-rate. The effect on the BE flow is severe, a major delay is introduced as a reaction to the increased rate of the AF flow. A slight increased in overall delay can be seen for the AF flow itself as well, but when compared to the BE flow the degradation is minimal. The effect of the increased AF rate on the EF classified flow is next to none.

When the EF ToS classified source increases its data-rate in the interval from 13 to 16 seconds, the BE flow again experience a severe increase in overall delay. The AF flow also experiences a small increase in delay, but again the increase is very small when compared to the BE flow. The EF classified flow experience little or none degradation when considering average delay, though some spikes in maximum delay are introduced as a result of the increased probability of congestion when increasing the data-rate.

In Figure 4.3b the maximum delay of the different flows are plotted. It is clear that the lowest prioritized BE flow must pay a high price for the increased performances experienced by the two other flows. The delay introduced for the BE flow when increasing the data-rates of either of the flows in the interval from 5 to 16 seconds is severe when compared to the two other flows.

Starting at 20 seconds, the available SE output bandwidth is reduced to 150×10^6 bits per second in the interval from 20 to 25 seconds, while a even narrower bandwidth, 130×10^6 bits per seconds, is applied in the interval from 25 to 28 seconds. The BE ToS classified flow experience a dramatic increase in delay, while the AF flow only experience a moderate increase in delay and the EF flow is degraded even less. The BE flow again pays the heaviest price of the reduction of resources available, as it is the lowest prioritized flow. Note that of the different parameter alterations applied in the SP simulation scenarios, the reduction of SE output bandwidth is the one which affects the higher prioritized data-flows the most.
A reduction to smaller packet-sizes transmitted from different sources is applied in the interval



(a) EF-flow

(b) Maximum delay

Figure 4.3: **Delay statistics of the EF classified flow and the maximum delays from each of the different classified data-flows.**

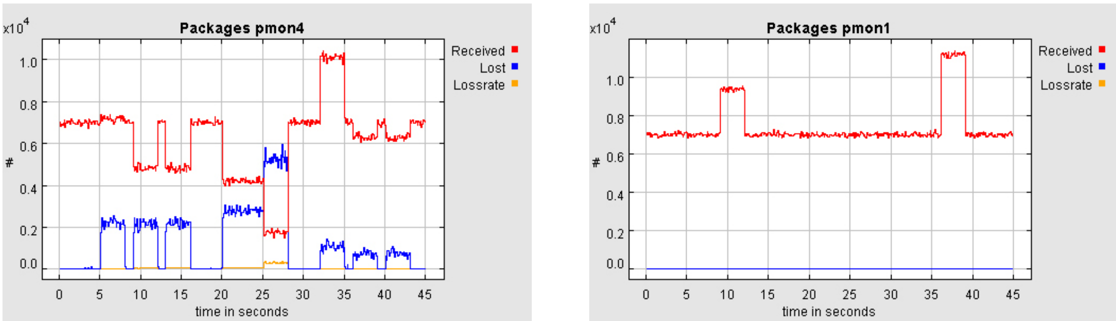
from 32 to 43 seconds. The first source that reduces the size of its packets is the BE ToS classified source. This occur in the interval from 32 to 35 seconds. The impact on the BE flow itself when considering delay is an increase in overall and maximum delay. For the two higher prioritized flows, the AF and the EF classified flows, little or none impact of the decreased packet size in the BE flow can be distinguishable. The packet-size of the BE flow is increased to its initial value at 35 seconds.

In the interval from 36 to 39 seconds the AF classified flow decreases its packet-sizes. The impact on the BE flow is an increased delay, while the effect on the AF flow itself is very small. An increase in maximum delay can be observed, but the increase in average delay is next to none. The introduction of maximum delay spikes is due to the increased packet rate which increases the probability of congestion, but the effect is not severe as the average is not increased distinguishably. For the EF flow the introduction of smaller packets in the AF flow has no effect on the delay performance.

Finally, the EF classified source decreases its packet size. Again the BE flow must pay the price, a delay similar to which was the case for the AF decreased packet size scenario is introduced. Neither the AF flow or the EF flow itself experience any increase in delay due to the decreased packet-sizes from the EF source.

In figure 4.3b the maximum delay values of each of the different classified sources are plotted. The plot clearly tells the characteristics of the SP approach when considering delay, the prioritizing enhances the performance of the higher prioritized flows, while the lower prioritized flows experience a significant degradation in delay performance.

4.2.2 Throughput Considerations



(a) BE-flow (b) AF-flow

Figure 4.4: **Received packets at the receiver, loss-rate and packets lost during transmission for the BE classified flow and the AF classified flow.**

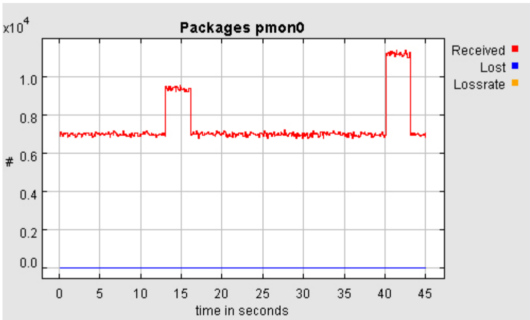
In figures 4.4a, 4.4b and 4.5a the packet-counts and loss-rates are shown for the BE classified flow, the AF classified flow and the EF classified flow respectfully. As the available bandwidth out from the SE is higher than the sum of the data-rates of the three incoming data-flows, there is little congestion in the SE node when initial parameters are applied.

Following the same time-line as for the delay considerations mentioned above, the data-rate of the BE flow is increased in the interval from 5 to 8 seconds. This also increased the received packet count for the BE classified flow, although more packets are lost. The increased BE flow data-rate has no impact on the packet count for neither of the two other flows.

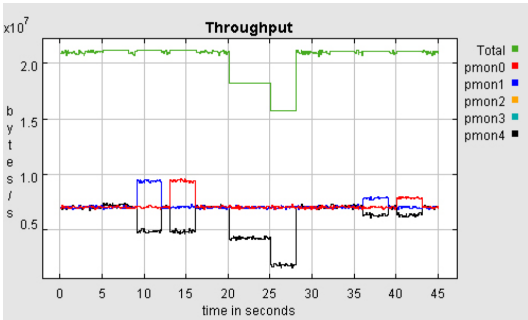
The AF source increases its data-rate in the interval from 9 to 12 seconds. The impact on the BE flow is severe, significantly fever packets are received and the loss-rate is increased. For the AF flow itself, the increased data-rate increases the received packet count without introducing a increase in packets lost. The EF flow is untouched by the alterations in the AF flow.

The increased data-rate of the EF source in the interval from 13 to 16 seconds increases the received packet count of the EF flow itself, while no impact at all is observed for the AF flow. Again the BE flow must pay the price in form of a reduced received packet count and an increased loss-rate.

The reduced SE output bandwidth introduced in the interval from 20 to 28 seconds has a severe impact on the BE flow. In the first interval (20-25 seconds), where the bandwidth is reduced moderately, the BE flow experience a significant reduction in packets received. Since the output rate of the BE source is left unaltered at its initial value, this means that the loss-rate increased. For the two other data-flows, there is impossible to observe any distinguishable degradation when considering packets lost or packets received. In the seconds interval with reduced SE bandwidth, from 25 to 28 seconds, the impact on the BE flow is even more severe. The packets lost count exceeds the packets received count, almost killing the BE flow completely off. The impact on the two other flows is again next to none. Again, the lowest prioritized BE flow pays the price for the stable performance of the two other flows.



(a) EF-flow



(b) Throughput

Figure 4.5: **Received packets at the receiver, loss-rate and packets lost during transmission for the EF classified flow. At the right the total system throughput is plotted.**

The last part of the simulation includes the packet size alterations made in the interval from 32 to 43 seconds. The first source to reduce its packet-sizes was the BE ToS classified source. This occurs in the interval from 32 to 35 seconds. The impact of this alteration on the BE flow itself is as seen in Figure 4.4a, both an increase in received packet count and an increase in the lost packet count. It has to be taken into account that the an increase in packet count while reducing packet-sizes does not automatically correspond to an increase in byte per seconds throughput, as will be mentioned further below. When considering the impact of the reduced BE packet-sizes on the two other flows, there is literally no impact at all when considering packet count or lost packets count.

In the interval from 36 to 39 seconds the AF classified source reduces its packet-sizes. This yields an increase in packet-count for the AF source, while in contrast to which was the case for the BE reduced scenario, no increase in lost packets count is observed. The BE flow must pay for the increases throughput of AF packets, an increase in packets lost and reduced amount of packets received can be observed for the BE flow. For the EF flow there is no impact at all.

In the final interval, from 40 to 43 seconds, the EF classified source decreases its packet-sizes. Similar to which were the case for the reduced AF packet size scenario, an increase in packets lost and a decrease in packets received can be observed for the BE flow. For the AF flow no differences from the initial setup can be observed, while the received packet-count of the EF flow has increased in the given interval.

In Figure 4.5b the throughput of the different classified flows are plotted together with the total system throughput. The result shows that the BE flow, when allowed, exploits any available resources as long as it does not affect the higher prioritized flows. This effect can be observed in the intervals 5 to 8 seconds and 32 to 35 seconds. A small increase in the total system throughput indicates that the BE flow uses the available resources best-effort up to the upper bound theoretical limit set by the total capacity of the system, without reducing the performance of the two higher prioritized flows. For the AF flow, with active alterations in the intervals 9 to 12 seconds and 36 to 39 seconds, a more greedy approach is shown. The AF classified flow enhances its own throughput performance at the cost of the BE performance. An identical scenario can be observed for the EF flow, which has its parameters changed in the intervals from 13 to 16 seconds and 40 to 43 seconds.

The main characteristics of the SP approach is that it can enhance the performance of higher prioritized data-flows at the cost of lower performance for other lower prioritized flows. The lowest prioritized flow works as a buffer of resources for the higher prioritized flows, a set of resources which can be exploited when needed due to an increased bit rate requirement or reduced link capacity. However, remembering the delay characteristics mentioned above, where the only massive degradation in delay for either of the two highest prioritized sources was observed when the bandwidth of the SE output was reduced severely, the lowest prioritized flow has of course a final amount of resources initially, and the degradation of performance spreads upwards in the prioritizing hierarchy when resources get spares. Another characteristic is the lack of lower bound performance protection for any of the flows, the resources of the BE flow could be completely exhausted when experiencing certain scenarios. This will kill the data-flow off completely, an effect which might be unwanted in some traffic policies.

4.3 Further Readings

In the following section a collection of literature where SP can be studied further is given. In [37] the basics of the class-based admission control scheme of SP scheduling are described. An average-case analysis together with a worst-case analysis of SP is given in [36]. In [38] an approach where priority scheduling is used in combination with other scheduling algorithms in emergency networks is studied. SP is furthermore discussed together with other priority schedulers with respect to QoS requirements in [39]. A further study of priority scheduling with focus on the phenomena of traffic self-similarity is given in [40].

4.4 Preliminary Conclusions

SP is an algorithms that offers great differentiation capabilities. The highest prioritized EF flow performs without flaw for most scenarios, the only exception is the reduced SE output bandwidth where the EF flow experience the a slight increase in average delay. The throughput of the highest prioritized flow is left untouched for all simulated scenarios, it is not dependent at all of the behavior of the other flows. In contrast, the lowest prioritized flow, BE, pays the price for the extremely stable performance of the highest prioritized flow. Increased data-rate of either of the flows, reduced SE output bandwidth, reduced packet-sizes, the impact of all alterations which would degrade the performances of the flows is handled by the lowest prioritized flow. The SP algorithm dictates furthermore that the lowest prioritized flow is to be exhausted for all resources before a higher prioritized flow is to experience a reduced performance. The SP algorithms capabilities of ensuring the performance of prioritized flows could be attractive in certain scenarios, but the potentially extreme unfairness without any lower-bound thresholds ensuring some QoS requirements of lower prioritized flows makes it an algorithm most suitable for specialized traffic policies.

Chapter 5

Deficit Round-Robin

The Deficit Round-Robin (DRR) scheduling approach is a queuing algorithm which divides the different data-flows into FIFO drop-tail sub-queues and dequeues from these respective queues in an iterative manner. During each iteration the DRR enabled node uses variables which each are corresponding to the sum of the number of allowed bits to transmit and the number of deficit bits from last iteration. The deficit variables gives the DRR enable node a higher degree of fairness since it reduces the impact of different packet sizes from different sources.

5.1 Algorithm Description

The main features of the DRR algorithm is its simplicity and fairness. The low complexity yields a simple implementation and its fairness where each of the data-flows perform almost independent of one another makes it an attractive scheduling algorithm. The most notably downside to the algorithm is its lack of differentiation between flows, all flows are treated the same. One could argue that this means that it lacks QoS support, but the fairness and thereby guaranteed allocation of parts of the available bandwidth is an important requirement for some services.

Consider i different data-flows arriving at the DRR enabled SE. Each of the i flows are routed onto i different queues, where each queue is denoted Q_i. The default amount of bits each flow i has available when routed onto queue Q_i is denoted by Q_i^N. The capacity of a single queue Q_i is C_i. The total queue capacity of the SE node is then $C = \sum_i C_i$. When transmitting from a given queue, the amount of bits transmitted at iteration k of the round-robin cycle is denoted $b_i(k)$. Transmission or dequeuing from a given queue is only allowed if $b_i(k) \leq Q_i^A(k)$, where $Q_i^A(k)$ is the allowance of the i'th queue in the k'th iteration. where Since the SE only transmits complete packets, no partial packets are transmitted. A given queue must wait until it has a sufficient $Q_i^A(k)$ for that at least one complete packet can be transmitted. A variable $D_i(k)$ is defined as the difference between the allocated bits of a given queue Q_i^N and the amount of transmitted bits $b_i(k)$:

$$D_i(k) = (Q_i^A(k) - b_i(k)). \tag{5.1.1}$$

The variable $D_i(k)$, which is the amount of deficit bits not used, is updated for each iteration of the round-robin cycle. In the conventional DRR approach the allocated bits for each flow, Q_i^N,

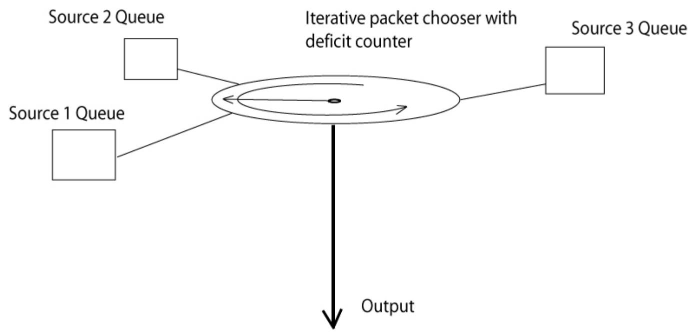


Figure 5.1: **The figure illustrates the characteristics of the DRR scheduling algorithm. The three different queues corresponding to the the three different data-flows are handled in an iterative manner dependent of the deficit counter variable of each of the respective queues.**

are initially set equal for all sources, while other suggested mutations of the approach implements differentiation between different flows. In this section an equal static bit allowance is used,

$$Q_i^N = Q_j^N \qquad \forall i, j. \tag{5.1.2}$$

After transmission of the $b_i(k)$ mentioned bits, the available capacity measure at the $k+1$ iteration is the sum of the default static allowance Q_i^N and the deficit of the previous iteration $D_i(k)$, yielding the expression

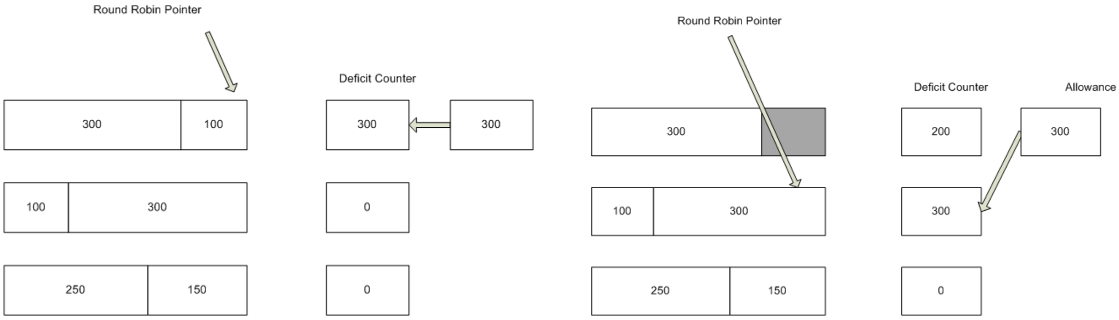
$$Q_i^A(k) = D_i(k-1) + Q_i^N. \tag{5.1.3}$$

Before a new transmission from the given queue can be allowed the expression

$$b_i(k) \leq Q_i^A(k) \tag{5.1.4}$$

must be true. A partial iteration is described in Figure 5.2. In the figure the initial deficit counter is set to zero for all flows. When the Round Robin Pointer moves to the first queue, the static allowance 300 is added to the deficit counter. The first packet in the first flow is 100 units large, which means that it fulfills the requirement stated by Equation 5.1.4 and can therefore be transmitted. Since the packet is only 100 units large, the deficit 200 units are reserved for the next iteration. This will leave an effective allowance for the first queue of 500 units in the next iteration since 200 + 300 = 500. In Figure 5.2b the Round-Robin Pointer has moved on step further in the cycle, pointing on the second queue. The packet at the front is 300 units large and can therefore be transmitted since it also fulfills the requirements of Equation 5.1.4. Since both the allowance and the packet-size is 300 units, there are no deficit bits which can be used in the next iteration, leaving the allowance for the next iteration at the default allowance of 300 units.
In the simulations in this thesis the different queues never run empty, but when considering other scenarios a queue could very well run empty. To avoid spending a considerable amount of time calculating and checking empty queues, an ActiveList which contains information concerning which queues that are active and which are not has been suggested.

The complexity of the algorithm can be expressed as follows. Consider i flows with maximum



(a) Initial setup (b) After transmission of the first queue

Figure 5.2: **The Deficit Counter is first set to its initial allowance before checking if transmission is allowed. The packet-size is smaller than the maximum allowed packet-size, and therefor the packet is transmitted. Since the packet is only 100 units the remaining deficit 200 units are saved/reserved for the next iteration.**

packet size Max. An incoming packet is routed onto its corresponding queue and appended to the tail of the queue. Both finding the correct queue and appending the packet requires $O(1)$ time complexity. When deploying the mentioned ActiveList approach, the dequeuing process only requires a constant number of operations in order to update the Deficit Counter and the ActiveList. If the the allowance of a flow Q is larger or equal to the maximum packet size Max, a packet will be dequeued for every visit to every queue, leaving the worst-case time complexity to $O(1)$ [4]. An illustrative implementation of the DRR approach in pseudo-code when handling 3 flows without use of the ActiveList approach:

```
create FIFOQueue Q1, Q2, Q3
create list DC
int Q

void initDC()
for (i=0, i<3, i++)
DC(i)=0;

void enqueue(Packet pkt)
i = pkt.FlowNumber
   if (i == 0)
   Q1.append(pkt)
   if (i == 1)
   Q2.append(pkt)
   if (i == 2)
   Q3.append(pkt)

void dequeue()
    if(Q1.isempty() == false)
      DC(0) = Q+DC(0);
```

```
  if(Q1.firstPkt().size <= DC(0))
   send(Q1.firstPkt())
if(Q2.isempty() == false)
  DC(1) = Q+DC(1);
  if(Q2.firstPkt().size <= DC(1))
   send(Q2.firstPkt())
if(Q3.isempty() == false)
  DC(2) = Q+DC(2);
  if(Q3.firstPkt().size <= DC(2))
   send(Q3.firstPkt())
```

5.2 Simulations

The main parameters are presented in Table 5.1, where only points along the time-line where alteration of parameters are made is listed. The interval between two mentioned time-line points are to be considered having static parameters within the given interval. Some of main parameters, such as the bandwidth of the links between the different sources and the SE, which are not manipulated throughout the simulation are either not mentioned at all or only marked by "-" in Table 5.1.

Time [sec]	Node	Data-rate [Mbps]	Bandwidth [Mbps]	Packet-size [bytes]
0	BE-source	44.8	-	80
0	AF-source	44.8	-	80
0	EF-source	44.8	-	80
0	SE	-	175	-
5	BE-source	**60**	-	80
8	BE-source	**44.8**	-	80
9	AF-source	**60**	-	80
12	AF-source	**44.8**	-	80
13	EF-source	**60**	-	80
16	EF-source	**44.8**	-	80
20	SE	-	**150**	-
25	SE	-	**130**	-
28	SE	-	**175**	-
32	BE-source	44.8	-	**50**
35	BE-source	44.8	-	**80**
36	AF-source	44.8	-	**50**
39	AF-source	44.8	-	**80**
40	EF-source	44.8	-	**50**
43	EF-source	44.8	-	**80**

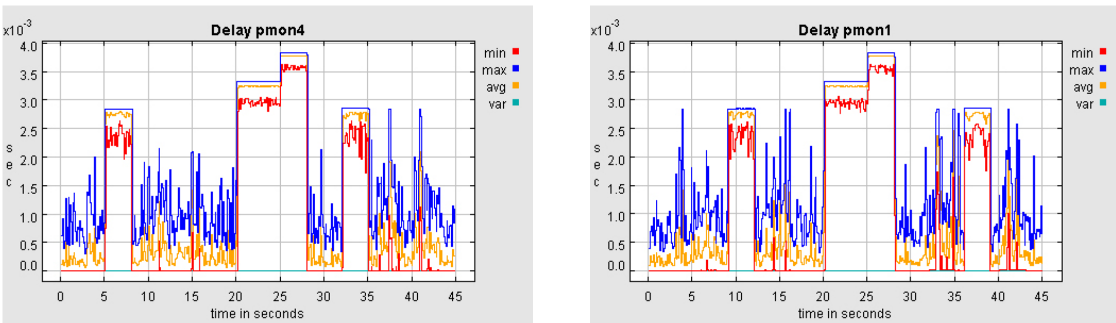
Table 5.1: Parameter alterations during the simulation

The sources have all Poisson distributed data-rates with individual starting seed, yielding some variation between the different sources, though with identical average rate. For more information concerning the simulation time-line, see Section 2.3. The capacity of each of the FIFO drop-tail queues within the SE is 20 kB, adding up to a total of 60 kB for all three queues. The total memory used for queuing when simulating this algorithm is thus identical to the total memory used when simulating the other algorithms in this thesis.

5.2.1 Delay Consideration

In figures 5.3a, 5.3b and 5.4a the different delay characteristic for each of the flows are plotted. Identically to the scenario used in the simulations in sections 3.3 and 4.2, some main parameters are altered throughout the simulation run. Before any alteration is applied, from 0 to 5 seconds, the system works with initial parameters, defining a normalized benchmark. The first part of the simulations where parameters actively are altered spans from 5 to 16 seconds. In this part the data-rates of the different sources are increased. As seen in Table 5.1, the first source to increase its data-rate is the BE classified source. Starting at 5 seconds, the BE source increases its rate to 60 Mbps. This data-rate is kept until 8 seconds of the simulation run, before the initial 44.9 Mbps is applied again.

When considering delay only, this has a significant impact on the BE flow itself. As seen in Figure 5.3a, the overall delay increases significantly. Both the average delay and the minimum delay values increase very much. The maximum delay also increases, but less distinguishably since there are also several quite high spikes when deploying the initial setup from 0 to 5 seconds. Another effect is that the overall delay is more stable, the variance is reduced when the increasing the data-rate. The impact of increasing the data-rate of the BE source on the two other data-flows (Fig. 5.3b, Fig.5.4a) is not very significant, although some spikes in the minimum delay occurs within the mentioned interval from 5 to 8 seconds. After 9 seconds, the



(a) BE-flow (b) AF-flow

Figure 5.3: **Delay statistics of the BE classified data-flow and of the AF classified data-flow.**

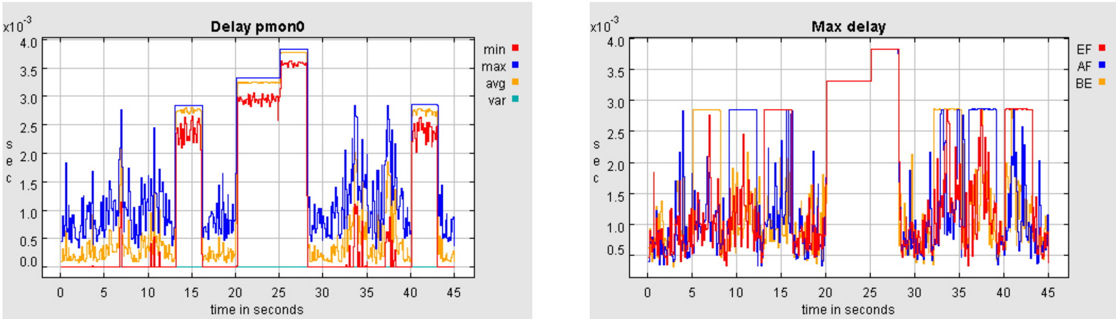
data-rate of the AF ToS classified source is increased to 60 Mbps. The increased data-rate is sustained in the following 3 seconds before being reduced to the initial 44.8 Mbps rate when the simulation reaches 12 seconds. The impact of increased data-rate on the AF flow itself is, similar to which were the case for the increased BE rate mentioned above, a significant increase in overall delay for the AF flow itself, seen in Figure 5.3b. The maximum delay, the average delay and the minimum delay are all significantly greater than what were the case when using initial parameters. Another effect of the increased rate is the stabilization of the delay, though at much higher values, but still there are less fluctuations and spikes in the AF delay characteristics when increasing the rate of the AF flow itself. The impact on the two other flows, the BE flow and the EF flow is small, only a slight increase in overall delay can be observed.

Similar to the two previous scenarios, the impact of an increased data-rate from the EF classified source has significant impact on the delay of the EF flow. Starting at 13 seconds the EF source transmits at 60 Mbps, before reducing it to the initial 44.8 Mbps after 16 seconds. As seen in Figure 5.4a, there is a significant increase in overall delay for the EF flow after 13 seconds. The increase stabilizes, again more stable than the highly fluctuating delay characteristics when using the initial parameters, at much greater values than what were the case when applying the initial parameter. The BE flow and the AF flow experience a slight increase in overall delay.

In Figure 5.4b the maximum delays of each of the flows are plotted. In the mentioned interval, from 5 to 16 seconds, it can be observed that the three different sources experience same delay characteristics when deploying higher rates. When altering the data-rate parameter of one source to a higher rate, a significant increase in delay is experienced by the corresponding data-flow, while the two other untampered flows only experience a slight increase in maximum delay.

The second main part of the simulation spans from 20 to 28 seconds. As described in Table 5.1, the output bandwidth of the SE is reduced, leaving less available resources in the shared link from the SE node to the respective receivers. The first alteration is done after 20 seconds, reducing the available bandwidth from 175 Mbps to 150 Mbps. This configuration is then held for 5 seconds, toward 25 seconds of the total simulation run. The impact of this alteration is similar for all of the flows (Fig. 5.3a, Fig. 5.3b, Fig. 5.4a), an identically high increase in overall delay can be observed for all flows. Similar to what were the case when increasing the data-rate of each of the different sources, a reduced SE output bandwidth makes the delay characteristics of each of the different flows more stable, there are less spikes and fluctuations in the maximum delay values, the average delay values and the minimum delay values.

After 25 seconds of the simulation run the SE output bandwidth is decreased even more, from 150 Mbps to 130 Mbps, before being increased to the initial value of 175 Mbps again after 28 seconds. The impact of this alteration is again mutual for all of the flows; a shared increase in delay can be observed. A further stabilization of the delay characteristics is again present, the average delay is more or less constant over the mentioned interval, in contrast to the extremely variating values when using the initial parameters. In Figure 5.4b the maximum delay values of each of the flows are presented, the shared characteristics can clearly be observed, all of the flows share an identical significant increase in maximum delay when the bandwidth of the SE node is reduced. As mentioned, the simulations parameters are set to their initial values at 28 seconds, and are not altered until 32 seconds where the final part of the simulation starts.



(a) EF-flow

(b) Maximum delay

Figure 5.4: **Delay statistics of the EF classified flow and the maximum delays from each of the different classified data-flows.**

In the final part of the simulation, spanning from 32 seconds to 43 seconds, the packet-sizes of the different flows are reduced. Starting at 32 seconds, the BE source transmits with a packet-size of 50 bytes, in contrast to the initial packet-size of 80 bytes. This setup is then used until 35 seconds, where the packet-size is increased to the initial parameter of 80 bytes per packet. As described in Figure 5.3a, the overall delay of the BE flow itself increases significantly when reducing the packet-sizes. The impact is more or less identical to the impact of the increased data-rate scenario mentioned above. A much greater delay and a stabilization of all delay characteristics can be observed. The two other flows (Fig. 5.3b, Fig. 5.4a), which transmit with initial parameters, experience a slight increase overall delay.

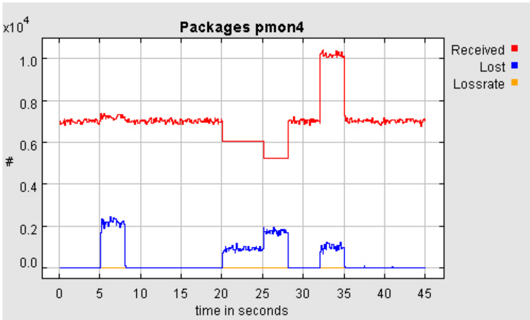
The AF source then reduces its packet-sizes to 50 bytes per packet after 36 seconds. This setup is used for 3 more seconds, before being altered back to its initial parameters of 80 bytes per packet at 39 seconds. The effects of this alterations are identical to the effects of the reduced BE packet-size scenario; a significant increase and stabilization of delay characteristics of the altered flow itself and a small increase in overall delay of the two unaltered flows.

After 40 seconds the EF classified source reduces its packet-sizes to 50 bytes per packet. A steep increase in delay can again be observed for the altered flow, while the two other flows experience only a minor increase in overall delay. The packet-size parameters is increased to its initial value of 80 bytes per packet after 43 seconds. In the interval from 43 to 45 seconds all of the system parameters are back to their initial values.

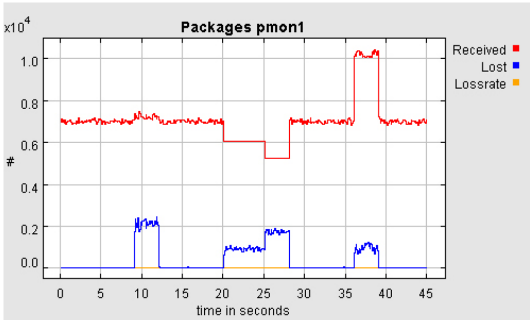
The mentioned impacts of the reduced packet-size alteration can be observed in the plot of the maximum delay values in Figure 5.4b, where a flow experience a significant increases in overall delay when it is reducing its packet-size. The mentioned increase in both maximum and overall delay for sources transmitting with initial packet-sizes in an environment where a reduced packet-size source transmits can also be observed.

5.2.2 Throughput Considerations

The packet-count of each of the different sources are displayed in figures 5.5a, 5.5b and 5.6a. When considering throughput the simulation time-line can, similarly to what were the case for the delay consideration, be divided into three main parts. The first part is where the data-rates of each of the flows are increased, the second part where the SE output bandwidth is reduced and the final part where the packet-sizes of the different sources are reduced.



(a) BE-flow



(b) AF-flow

Figure 5.5: **Received packets at the receiver, loss-rate and packets lost during transmission for the BE classified flow and the AF classified flow.**

Starting at 5 seconds the BE source increases its data-rate from 44.8 Mbps to 60 Mbps. 3 seconds later the BE source reduces its rate to 44.8 Mbps again. The impact of this alteration on the BE flow itself (Fig. 5.5a) is a small increase in the received packet count, while the lost packet count increases significantly. This is due to the a higher probability of overflow in the final capacity FIFO drop-tail queue. The impact on the two other flows (Fig. 5.5b, Fig. 5.6a) is undistinguishable when considering packet-count.

After 9 seconds the AF source increases its data-rate to 60 Mbps. This rate is kept until 12 seconds, when the rate is reduced to the initial 44.8 Mbps again. The effect of this alterations is similar to the effects of the BE rate alteration; a slight increase in the received packet count and a significant increase in lost packet count for the altered flow itself. The two other flows, in this case the BE flow and the EF flow, experience no distinguishable difference from the initial packet count characteristics.

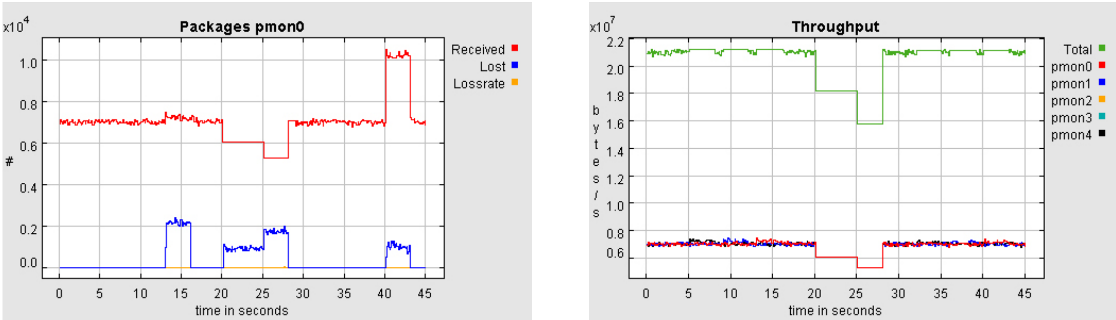
In the interval from 13 seconds to 16 seconds the EF source increases its data-rate from 44.8 Mbps to 60 Mbps. Again it is only the altered flow that exhibits any difference in packet-count characteristics; a slight increase in packet received count and a significant increase in packets lost count.

When considering the total system throughput in Figure 5.6b it can be observed that an increase in data-rate for either of the flows induce a slight increase in total system throughput. The difference is corresponding to the slight increase in received packet count for each of the different flows when using an increased rate. The increase in system throughput is limited by the available bandwidth at the output of the SE node. Another issue is that the packet-count characteristics of the flows are independent of the characteristics of one another, yielding quite fair performances. The increase in data-rate of one flow does not reduce the received packet count of another flow. This result is in contrast to the results from the delay considerations in Section 5.2.1, where an increase in data-rate of one flow induced a slight increase in delay for the two other flows, and thereby not exhibiting independent characteristics with respect to data-rate.

After using initial parameters in the interval from 16 to 20 seconds, the SE output bandwidth is reduces from 175 Mbps to 150 Mbps in the interval from 20 seconds to 25 seconds. The impact of this alteration is identical for every flow, the received packet-count is significantly reduced and the lost packet count is correspondingly increased. When reducing the available bandwidth even more, to 130 Mbps in the interval from 25 seconds to 28 seconds, the received packet count is again reduced and the packet lost count is increase. The increase in lost packets when reducing the available resources is due to the increased probability of overflow in the final capacity FIFO drop-tail queues used by all flows within the SE when packages arrive faster than they depart. The total system throughput is, as shown in Figure 5.6b, reduced correspondingly to the reduction in received packet count of each of the different flows. The SE bandwidth is increased to its initial value after 28 seconds, and all the different parameters are set to their initial value in the interval from 28 seconds to 32 seconds.

In the final part of the simulation run the packet-sizes used by the different sources are reduced. Starting at 32 seconds the BE source decreases the packet-size it is using to 50 bytes per packet. It can be observed in Figure 5.5a that this induces an increase in packet received count, at the same time as there is also an increase in the packet lost count. Note that the increase in packet received count does not automatically mean that the throughput is increased since the packets are smaller. The impact on the two other flows is none, no difference in either the packet received count or the packet lost count can be observed. The packet-sizes of the BE flow are increased to their initial value of 80 bytes per packet after 35 seconds.
In the interval from 36 to 39 seconds the AF source reduces its packet-size to 50 bytes. Identical to the reduced BE packet-size scenario, the only observable difference is on the AF flow itself, which exhibit an increase in both the received packet count and the lost packet count. After 39 seconds the packet-size is again set to its initial value of 80 bytes per packet. In the interval from 40 seconds to 43 seconds the EF source decreases its packet-size and again identical to the two previous mentioned scenarios, an increase in both the packet received count and the lost packet count can be observed. In the final interval from 43 seconds to 45 seconds, all the system parameters are again back to their initial values.

The different received packet counts for each of the different flows is independent of the packet-size used by any of the flows. A reduction in packet-size in one flow does not affect the throughput of any of the other flows, leaving the packet count characteristics independent of one another. This is again in contrast to the delay characteristics described in Section 5.2.1, where a reduction



(a) EF-flow

(b) Throughput

Figure 5.6: **Received packets at the receiver, loss-rate and packets lost during transmission for the EF classified flow. At the right the total system throughput is plotted.**

in packet-size in one flow induced an increase in delay for all of the other flows.

In Figure 5.6b the impact of reduced packet-size on the total system throughout can be observed. Consider multiple sources transmitting with a packet-size of one bit. For an allowance for every queue larger than one bit, at least one packet would be transmitted per round-robin cycle from every queue. This means that the instant packet throughput would increase when compared to a scenario where larger packets were used. Consider a scenario where only one of multiple sources transmit with a packet-size of one bit. The source using one bit packets would transmit in every iteration while the others would have to wait for the sum of the allowance and the deficit counter to grow large enough for a whole packet. Also, the probability for that the sum of the deficit counter and the allowance to be exact the size of the waiting packet is smaller with larger packets than for the one bit packet flow where every sum of the allowance and the deficit counter will be an exact number of packets. This yields an increase in throughput for the source transmitting with smaller packets. As seen in Figure 5.6b, the total system throughput reaches its theoretical limit set by the SE output bandwidth when reducing the packet-size of any of the sources. Also, the characteristic of one flow not dictating the performance of the other flows, leaving fair scheduling, is attractive.

5.3 Further Readings

In the following section a collection of literature where DRR can be studied further is given. A further introduction to DRR is given in [4], while effects of using different default Q_i^N bit allowances is studied in [41]. In [42] the the latency characteristics of DRR is described, while a use of DRR algorithms in IEEE 802.11e Wireless LAN is described in [43]. In [44] a version of DRR dedicated to cellular wireless communications is studied.

5.4 Preliminary Conclusions

DRR offers fair scheduling based on the iterative Round-Robin approach. The different simulation scenarios shows that, when considering delay, alterations in either data-rate or packet-size in one of the flows, the altered flow experience the major increase in delay whilst the other flows only experience a slight increase. When considering the reduced SE output bandwidth scenario, all of the flows experience identical performance degradation. Furthermore, when considering throughput, an increase in data-rate or reduction of packet-size of one flow has no impact on the remaining flows, it only affects the altered flow itself. The reduced bandwidth scenario affects all of the flows identically. The different mentioned characteristics makes DRR robust against ill behaving sources, the behavior of one flow affects the performance of the other flow very little. The robustness against reduced packet-sizes is due to the deficit counter mechanism which ensures that, in contrast to conventional Round-Robin algorithms, the performances of the different flows are independent of the packet-sizes. The increased data-rate scenario does not affect the performance of the unaltered flows due to the iterative nature of Round-Robin.

When considering QoS, DRR offers no differentiation between flows. The lack of differentiation could make it unsuitable in modern networking nodes, where differentiation between flows requiring different QoS is of major interest. Algorithms offering both differentiation and higher fairness (Chapter 2.6) have been proposed, but they all share the drawback of higher complexity. The low complexity of DRR makes it thus an interesting algorithm when considering high-speed networks.

Chapter 6

Deficit Weighted Round-Robin

The Deficit Weighted Round-Robin (DWRR) scheduling algorithm is an evolution of the conventional Deficit Round-Robin algorithm described in Chapter 5. Using DRR as a basis, the DWRR offers a new dimension to the scheduling when compared to the conventional DRR; differentiation between different data-flows can be made through simple weighting of each flow. The capability of differentiation between the sources makes the DWRR an interesting algorithm when considering scheduling of packets from different flows with different QoS requirements.

6.1 Algorithm Description

The DWRR algorithm is closely bounded to the DRR algorithm. Most of the main implementation features are shared, the only difference is the capability of weighting each flow with a certain constant. This weighting does, as will be described in more detail below, introduce the capability of differentiation between flows, a major factor in a network node offering what may be described as full QoS support.

Using the notation defined in Section 5.1, the DWRR enabled SE receives data from i different sources. The different flows are routed onto i different FIFO drop-tail queues within the SE. A major feature of both DRR and DWRR are their iterative characteristic. This means that the algorithm work at a cycle based work schedule. At each cycle the different queues get a certain amount of bits which they can transmit. This allowance is denoted by Q_i^N. When transmitting from a queue, the amount of transmitted bits in the k'th iteration is denoted $b_i(k)$. Note that the amount of transmitted bits must be equal to the size of either one or several packets, in other words, only whole packets are allowed for transmission. The expression

$$b_i(k) \leq Q_i^A(k) \tag{6.1.1}$$

where $Q_i^A(k)$ is the allowance for the i'th queue in the k'th iteration must be true if the given queue should be allowed to transmit. Since only complete packets are allowed, a scenario where the whole allowance is not used completely is very likely to occur. The weighted deficit amount of bits after a transmission is denoted by the variable $D_i(k)$, expressed by

$$D_i(k) = (Q_i^A(k) - b_i(k) + W_i - 1)/W_i \tag{6.1.2}$$

where $Q_i^A(k)$ is the allowance made for queue j in the k'th iteration, $b_i(k)$ is the amount of transmitted bits in the k'th iteration and W_i is the constant, preset weight for the i'th flow. When comparing this expression to the deficit variable defined in Section 5.1 by Equation 5.1.1, it is obvious that the only difference is the weighting factor. By setting the weight to 1 for all i flows yields Equation 6.1.2 equal to Equation 5.1.1.

The mentioned allowance Q_i^N could either be set static and equal for each flow or it could be different from one queue to another. In conventional DWRR the differentiation between different flows is done with respect to the individual queue weight W_i, and it is this approach which is used in this thesis. The allowance can be expressed by the equation

$$Q_i^N = Q_j^N \qquad \forall i, j. \tag{6.1.3}$$

which makes the standard allowance of every queue equal and not dependent of the iteration variable k. Of course the effective allowance for every queue is given by the sum of the standard allowance and the weighted amount of deficit bits, expressed by

$$Q_i^A(k) = Q_i^N + D_i(k-1) \tag{6.1.4}$$

where Q_i^N is the constant standard allowance and $D_i(k-1)$ is the weighted amount of deficit bits calculated in the previous iteration.

Since the main features of DWRR are similar to DRR, both Figure 5.1 and Figure 5.2 in Section 5.1 can be used for describing DWRR as well. The complexity of DRR is shared with DWRR, it requires only $O(1)$ time complexity. A implementation of the ActiveList, a list where only the active queues are to be present so that the SE does not need to spend resources on inactive queues, could also of course be made for DWRR. In the following simulations the ActiveList has not been implemented as the nature of the sources dictates that neither of the queues can run empty [4].

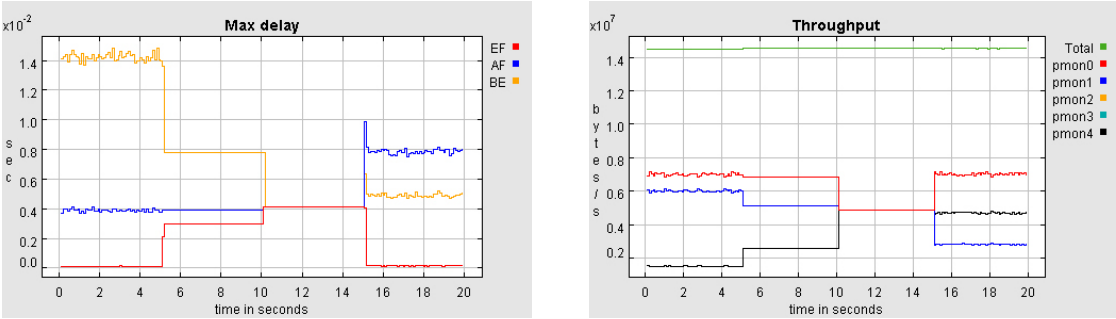
An illustrative implementation of the DWRR approach in pseudo-code when handling 3 flows without use of the ActiveList approach:

```
create FIFOQueue Q1, Q2, Q3
create list DC
int Q
create list W

void initDC()
for (i=0, i<3, i++)
DC(i)=0;

void setWeight(int queueNr, int weight)
W(queueNr)=weight;

void enqueue(Packet pkt)
```

```
i = pkt.FlowNumber
   if (i == 0)
   Q1.append(pkt)
   if (i == 1)
   Q2.append(pkt)
   if (i == 2)
   Q3.append(pkt)

void dequeue()
    if(Q1.isempty() == false)
      DC(0) = Q+(DC(0)*W(0)-W(0)+1);
      if(Q1.firstPkt().size <= DC(0))
       send(Q1.firstPkt())
    if(Q2.isempty() == false)
      DC(1) = Q+(DC(1)*W(1)-W(1)+1);
      if(Q2.firstPkt().size <= DC(1))
       send(Q2.firstPkt())
    if(Q3.isempty() == false)
      DC(2) = Q+(DC(2)*W(2)-W(2)+1);
      if(Q3.firstPkt().size <= DC(2))
       send(Q3.firstPkt())
```

6.2 Simulations

Before the main part of the simulations is presented, a short presentation of a simulation run that describes the impact of the weighting of each of the queues is given. The simulation run is 20 seconds long where three separate data-flows are designated different weights initially, before being altered several times throughout the simulation run. The maximum delay values of each of the queues are shown in Figure 6.1a, while Figure 6.1b shows both the throughput of all the different flows and the total system throughout. Note that the sources transmit at the same rate as previously in this these, 44.8 Mbps per source, while the SE output bandwidth has been reduced to 120 Mbps. The low bandwidth means that there is not enough bandwidth for all of the flows to perform without congestion over the shared link from the SE to the receivers.
Furthermore, the simulation time line is divided into several intervals where the weights are constant, starting with the interval from 0 seconds to 5 seconds. In this interval the BE flow weight is set to 1, the AF flow weight to 4 and the EF weight to 16. The actual implementation is in contrast to Equation 6.1.2 where it follows that a lower weight value yields higher bit allowance. During the mentioned first 5 seconds the EF flow is clearly prioritized highest, the AF flow second and the BE flow last. When considering the throughput of the BE flow, it is close to zero. This means that the BE flow is sacrificed for the performance of the two higher prioritized flows. A low prioritized flow can, when considering the SE handling isolated, never be killed off completely since the sum of the weighted amount of deficit bits and the basis allowance will grow steadily for each iteration. Sooner or later the allowance will grow large enough for a whole packet to be transmitted. This could though only be a theoretical characteristic of the DWRR



(a) Maximum delay

(b) System throughput

Figure 6.1: **Maximum delay and total system throughput when alternating the weights of the different flows.**

enabled SE since a data-flow could in practice be killed off due to high loss-rate or delay longer than the time-to-live duration for a given packet.

In the interval from 5 seconds to 10 seconds the flow weights are altered as follows; the EF weight is untouched, the AF weight is set to 12 and the BE weight to 6. The impact of this alteration is a more equal transmission environment for all flows, though the BE flow is still performing significantly poorer than the two others.

After 10 seconds the weights are set equally to 10 for all sources, leaving an identical transmission environment for all flows. This setup is used for 5 seconds, until a differentiation is again applied after 15 seconds. This time the EF flow weight is set to 60, the AF flow weight to 12 and the BE flow weight to 20. The impact is obvious, the EF flow performs perfectly, the AF flow quite poor, while the BE flow is now the second prioritized flow and performs more or less adequately taking the low bandwidth conditions into account.

The possibility of changing the flow weights to any arbitrary combination of weights yields a great degree of freedom for differentiation between the different flows when designing a SE. Flows could get a guaranteed part of the available bandwidth independent of the given transmission conditions, or flows could ultimately be sacrificed totally for the performance of other flows. When compared to the conventional DRR algorithm, this is a major advantage of the DWRR algorithm.

In the simulation scenario used in the following sections the flow weights are set constant as follows:

- BE-queue weight: 1
- AF-queue weight: 4
- EF-queue weight: 16

The main parameters are presented in Table 6.1, where only points along the time-line where alteration of parameters are made is listed. The interval between two mentioned time-line points are to be considered having static parameters within the given interval. Some of main parameters, such as the bandwidth of the links between the different sources and the SE, which are not manipulated throughout the simulation are either not mentioned at all or only marked by "-" in Table 6.1.

The sources have all Poisson distributed data-rates with individual starting seed, yielding some variation between the different sources, though with identical average rate. For more information concerning the simulation time-line, see Section 2.3. The capacity of each of the FIFO drop-tail queues within the SE is 20 kB, adding up to a total of 60 kB for all three queues. The total memory used for queuing when simulating this algorithm is thus identical to the total memory used when simulating the other algorithms in this thesis.

Time [sec]	Node	Data-rate [Mbps]	Bandwidth [Mbps]	Packet-size [bytes]
0	BE-source	44.8	-	80
0	AF-source	44.8	-	80
0	EF-source	44.8	-	80
0	SE	-	175	-
5	BE-source	**60**	-	80
8	BE-source	**44.8**	-	80
9	AF-source	**60**	-	80
12	AF-source	**44.8**	-	80
13	EF-source	**60**	-	80
16	EF-source	**44.8**	-	80
20	SE	-	**150**	-
25	SE	-	**130**	-
28	SE	-	**175**	-
32	BE-source	44.8	-	**50**
35	BE-source	44.8	-	**80**
36	AF-source	44.8	-	**50**
39	AF-source	44.8	-	**80**
40	EF-source	44.8	-	**50**
43	EF-source	44.8	-	**80**

Table 6.1: Parameter alterations during the simulation

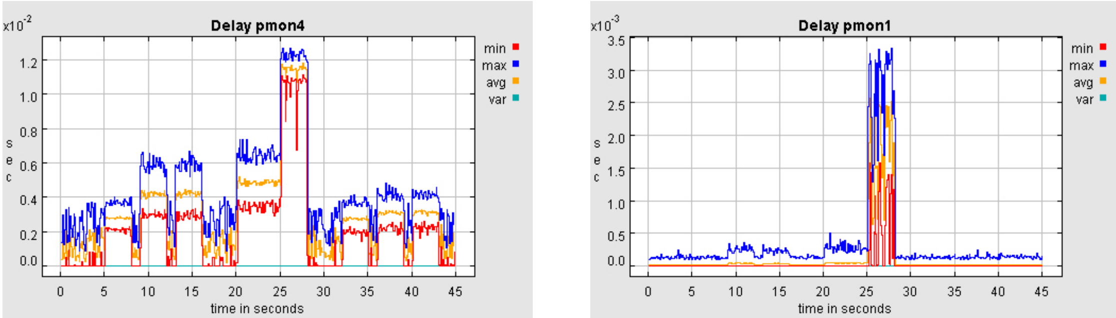
6.2.1 Delay Considerations

In Figure 6.2a, Figure 6.2b and Figure 6.3a the different delay characteristics of the different ToS classified flows are shown. As mentioned in the previous chapters and in Table 6.1, the simulation time-line can be divided into three main parts. The first part is from 0 to 16 seconds, where the focus is on what impact an increased data-rate of either of the flows has on the delay of the flows. The data-rate is altered in similar fashion as in the previous chapters, starting with the BE source. After 5 seconds of the simulation run the BE source increases its data-rate from the initial value of 44.8 Mbps to 60 Mbps. This change is the only one applied, all other system parameters are left untouched at their initial values. The increased BE data-rate is kept for 3 seconds before being reduced to its initial value of 44.8 Mbps after 8 seconds of the total simulation run. The impact of the increased data-rate on the BE flow itself can be observed in Figure 6.2a. The overall delay is clearly increased in the mentioned interval, the minimum delay values, the average delay values and the maximum delay values are all greater when increasing the data-rate. This comes as no surprise since the BE flow is the least prioritized flow in the system, the cost of the increased data-rate must solely be payed by the BE flow itself. The increased rate induces more congestion in the BE flow, and thus higher delay. There is also a stabilization of the delay characteristics when increasing the rate, less fluctuation and spikes in the delay values, but as for DRR, at much higher overall values than what were the case for the scenario using the initial rate. For the two other flows there is no difference in performance when comparing the increased BE data-rate scenario to the scenario using the initial BE data-rate.

The next interval where changes are applied is from 9 seconds to 12 seconds. In this interval the AF source increases it data-rate from 44.8 Mbps to 60 Mbps. The impact on the BE source is severe. A significant increase in overall delay can be observed in Figure 6.2a. Similar to the scenario with the increased BE rate, the BE delay characteristics are more stable when the AF source increases its rate, though with significantly greater values. The impact of the increased AF rate on the AF flow itself is a slight increase in delay. As can be observed in Figure 6.2b, both the average delay and the maximum delay gets a small but still distinguishable increase in delay. The effect of the increased AF rate on the EF flow is next to none, the delay performance is very modestly increased. Only a slight increase in average delay and maximum delay can be observed in Figure 6.3a. The AF source reduces its data-rate back to the initial value of 44.8 Mbps after 12 seconds of the total simulation time.

In the following interval from 13 seconds to 16 seconds, the EF source increases its data-rate from 44.8 Mbps to 60 Mbps. Starting with the BE flow, the impact of the increase in data-rate of the EF flow is significant. Again a great increase in overall delay can be observed, together with a stabilization of the delay characteristics. The EF flow also experience an increase in overall delay when compared to the scenario where the initial rate of 44.8 Mbps were applied by all three sources. Both the average delay and the maximum delay are increased slightly, but not as much as for the previous scenario where the AF source increased its rate itself. This implies that the DWRR algorithm has fairness characteristics which could be attractive in certain scenarios. The impact of the increased EF rate on the EF flow itself is a slight increase in delay. Greater than what were the case for the AF increased rate scenario, which again implies the potential fairness characteristics of DWRR. A small increase in average delay and a quite significant increase in maximum delay can be observed in Figure 6.3a.

In Figure 6.3b the maximum delay values of each of the data-flows are collected in one plot. The magnitude of the effect of the different weights for the different flows is immense. The impact



(a) BE-flow

(b) AF-flow

Figure 6.2: **Delay statistics of the BE classified data-flow and of the AF classified data-flow.**

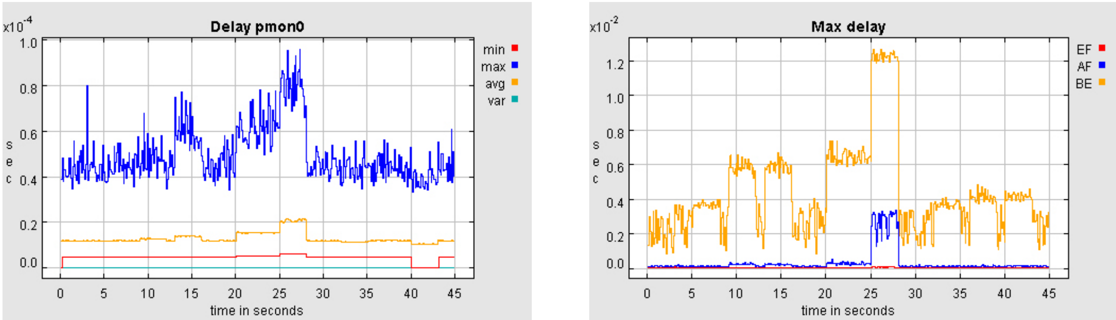
of increasing the data-rate of either of the flows are none for the EF flow and next to none for the AF flow when compared with immense impact on the BE flow.

In the following four seconds, from 16 seconds to 20 seconds of the simulation time, all the parameters are back at their initial values. Starting at 20 seconds the SE output bandwidth is reduced from 175 Mbps to 150 Mbps. The effect on the BE flow is again severe. It pays the price for the more or less unaffected delay performances of the two other flows, which only experience a slight increase in overall delay.

After 25 seconds the SE output bandwidth is reduced even more, from 150 Mbps to 130 Mbps. This bandwidth is kept in the following 3 seconds, before being increased to its initial value of 175 Mbps after 28 seconds. In this scenario its not only the BE flow that experiences a significant increase in delay, also the AF flow experience a severe increase in overall delay. The BE flow delay is increased to new heights, while keeping the stability of the delay characteristics. Since it is obvious that the BE resources are alone no longer sufficiently large enough to cope with the reduced bandwidth, a larger portion of the resources needed for keeping the EF performance stable must be taken from the AF resources. This materializes through a higher delay for the AF flow. The delay characteristics of the AF flow in the mentioned interval from 25 seconds to 28 seconds are dominated by spikes and rapid fluctuations. This implies that the amount of available resources lies at the borderline between sufficient enough for adequate AF delay performance and severely degraded AF delay performance. As previously implied, the impact of the further reduction of the SE output bandwidth only increases the overall delay of the EF flow slightly.

When considering the maximum delay distribution in Figure 6.3b, it can be observed that the BE flow again pays the price for more or less unaffected AF and EF delay characteristics in the interval from 20 to 25 seconds, while the resources available for distribution in the BE flow runs dry in the following interval from 25 to 28 seconds, leaving the AF flow also severely affected in the mentioned interval.

In the final part of the simulations the different sources reduces the packet-sizes one after another. Again starting with the BE classified flow, the BE source reduces its packet-sizes from 80



(a) EF-flow (b) Maximum delay

Figure 6.3: **Delay statistics of the EF classified flow and the maximum delays from each of the different classified data-flows.**

bytes to 50 bytes in the interval from 32 to 35 seconds. As seen in Figure 6.3a, this alteration increases the overall delay of the BE flow itself significantly. The average delay, the minimum delay and the maximum delay are all increased. A stabilization of the delay characteristics, similar to what can be observed when increasing the data-rates of either of the flows or reducing the SE output bandwidth, can be observed. The increase in delay can be explained by the increase in relative time spent in the queue when using smaller packets. The impact of reduced BE packet-size on the AF flow are undistinguishable when considering Figure 6.2b. For the EF flow there is a very small decrease and stabilization of the average delay, no other effect can be observed. After 35 seconds of the total simulation time, the BE source increase its packet-sizes to 80 bytes per packet again.

The next source to alter its packet-sizes is the AF source. Starting at 36 seconds it reduces its packet-sizes from 80 bytes per packet to 50 bytes per packet. This induces an increase in delay for the BE flow, again with reduction of the variance of the different delay measures. The AF flow experience a slight increase in peak maximum delay only, the average delay values and the minimum delay values are not increased distinguishably. The impact of reduced AF packet-size on the EF flow is none.

In the final interval where alterations are made the EF source reduces its packets. The effect of this alteration on the EF flow itself is a reduction of the overall delay. The minimum delay is significantly reduced since smaller packets would spend less time in the queue during a best case scenario where the queue allowance per iteration fits perfectly with the packet-sizes, a scenario which is more likely with smaller packets. There is also a reduction in maximum delay, thus leaving the average delay also reduced. The BE flow does again pay the price as it again experience a severe increase in overall delay. The AF flow is left unaffected. In the final 2 seconds of the simulation, from 43 seconds to 45 seconds, all the system parameters are back at their initial value.

When considering the maximum delay values of the different queues to one another in Figure 6.3b, it can be observed that the AF flow and the EF flow are left more or less unaffected of the packet-size alterations when compared to the BE flow. The BE flow must pay the price for the performance of the two other flows. This could imply a lack of support for fairness for the DWRR algorithm, but as mentioned before, the flow weights could easily be altered for more fairness between the different flows if that was the goal of a given traffic policy.

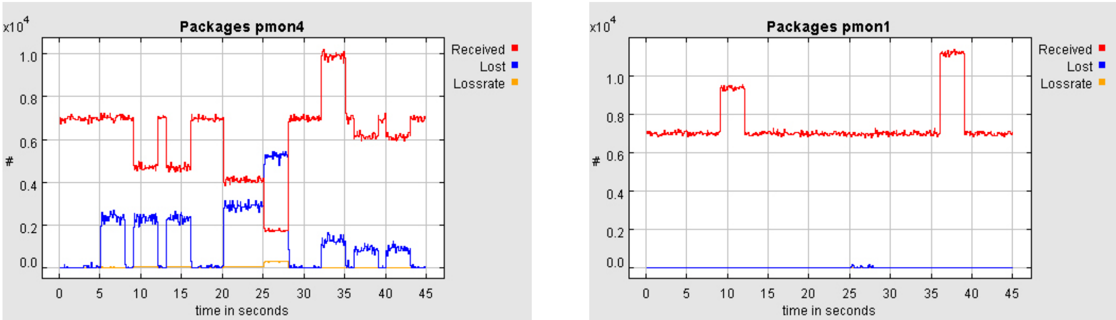
6.2.2 Throughput Considerations

Using the same simulation scenarios as for the delay considerations in the previous section, the packet-count of the different sources are displayed in figures 6.4a, 6.4b and 6.5a.

Starting with the increased BE data-rate in the interval from 5 seconds to 8 seconds, only a slight increase in received packet count can be observed for the BE flow itself. The main feature which dominates the BE flow when increasing its data-rate is the severe increase in packets lost. An increase in data-rate for the lowest prioritized flow, when transmitting in an environment which already before the increase in rate operates with almost exhausted capacity, yields a sharp increase in packets lost. The packet-count performances of the two higher prioritized flows, the AF classified flow and the EF classified flow, are not influenced by the increase of data-rate for the BE flow. As mentioned, the BE source transmits with an increased rate of 60 Mbps in 3 seconds, from 5 seconds of the simulation run until 8 seconds, before the rate is reduced again to its initial value of 44.8 Mbps.

In the next interval where a system parameter is altered, starting at 9 seconds and lasting till 12 seconds of the simulation run, the AF source increases its rate from the initial 44.8 Mbps to 60 Mbps. The impact of this increase on the AF flow itself can be observed in Figure 6.4b. There is a significant increase in received packet count for the AF flow itself when increasing its rate, while the packet lost count metric experience no difference from when transmitting with the initial rate, still no packets are lost. This is in contrast to the impact the increased AF rate has on the BE flow. The BE flow experience a severe decrease in performance when considering the throughput, a significant decrease in packets received and thus also a significant increase in packets lost is the result of the AF flows behavior. Since the SE already initially operates with almost exhausted capacity, an increase in data-rate for either of the sources would ultimately yield a decrease in performance for one or several of the flows. Since the BE flow has the lowest priority, its deficit amount of bits are weighted lowest, the BE flow must pay the price when the capacity of the SE becomes exhausted. As for the highest prioritized flow, the EF flow, the increase in AF rate has no influence on its performance.

The EF flow increases its rate in the interval from 13 to 16 seconds, from 44.8 Mbps to 60 Mbps. The impact of this increase on the EF flow itself is an increase in received packet count. The lost packet count is not influenced by the increase in data-rate, still no packets are lost when increasing the data-rate of the EF flow. No distinguishable effects of the EF increased rate can be observed on the AF flow, it operates as if no changes were made when compared to the initial setup. This is in great contrast to the performance of the BE flow. Similar to when the AF flow increased its rate, the BE flow must pay the price for the increased EF rate in form of degraded performance. A significant decrease in packet received count and a corresponding increase in



(a) BE-flow (b) AF-flow

Figure 6.4: **Received packets at the receiver, loss-rate and packets lost during transmission for the BE classified flow and the AF classified flow.**

packets lost is the result of the increased EF rate for the BE flow.

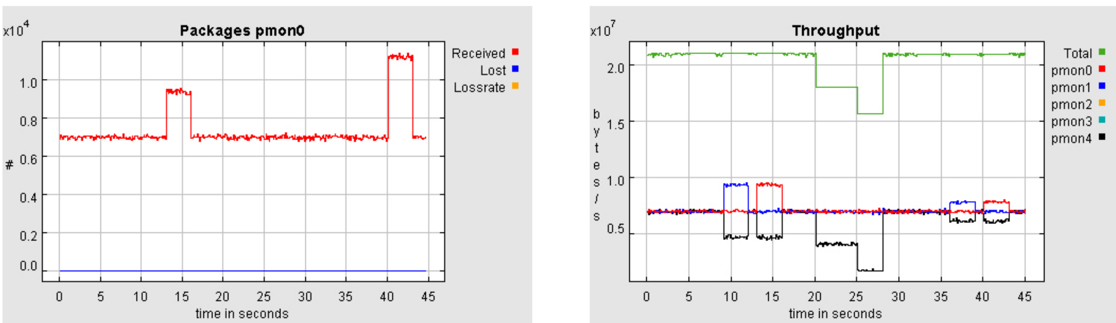
When considering 6.5b in the mentioned interval from 5 seconds to 16 seconds, it is evident that the performance of the BE flow is very dependent of the behavior of the two other flows. When increase its own rate it experience no increase in throughput, while when the two other sources increases their rates, they get a significant increase in performance while the BE flow gets its performance degraded. If the capacity of the SE were infinite, all the flows could in theory operate with perfect performance independent of the rate of one another. But since the capacity indeed is final, and almost exhausted when applying the initial parameters, an increase in rate for either of the sources will yield a degraded performance for the BE flow since it is prioritized lowest. The resources made available for the other flows from the BE flow are of course final, if the EF flow was to increase its rates even more and thereby totally exhausting the resources potentially made available from the BE flow completely, the second lowest flow, meaning the AF flow, would start to experience a degradation in its performance too. These characteristics could be argued as unfair, but as mentioned before, the weights of the flow could be tuned more fairly, leaving the burden of performance degradation more fairly distributed among the flows.

The next part of the simulation worth mentioning is the interval from 20 seconds to 28 seconds. Starting at 20 seconds the SE output bandwidth is decreased from 175 Mbps to 150 Mbps. This bandwidth is kept for the following 5 seconds, until it is changed again after 25 seconds of the total simulation run. The impact of this alteration on the BE flow is severe. The received packet count drops significantly, and correspondingly, the lost packet count is increased. The two other flows are left untouched by the change in SE output bandwidth. At 25 seconds the SE output bandwidth is decreased even further, from 150 Mbps to 130 Mbps. When operating with this amount of resources, the BE flow is almost completely killed off in practice. The amount of received packets is surpassed by the amount of lost packets, leaving the loss-rate disgraced. The reduction of resources available is so great that the BE flow alone no longer can cope with it alone, the reduction does also affect the higher prioritized AF flow. A slight increase in the lost packet count can be observed in the mentioned interval for the AF flow.

The performance of the EF flow is still untouched by the system alteration when considering the received packet count and the lost packet count. After 28 seconds of the simulation the SE output bandwidth is increased to its initial value of 175 Mbps, leaving all system parameters at their initial values for the next 4 seconds.

The throughput of the different flow is as mentioned show in Figure 6.5b. In the interval from 20 seconds to 25 seconds the total system throughput is reduced as a result of the reduction of available bandwidth. The lowest prioritized BE flow is as mentioned deprived from resources for the sake of the performance of the two higher prioritized flows. In the interval where the bandwidth is reduced further, from 25 seconds to 28 seconds, the BE flows throughput is reduced correspondingly to the bandwidth reduction. There is not a distinguishable reduction in throughput for any of the other flow in this interval either, but as mentioned above, the performance of the AF flow is reduced since it experience a slight increase in the lost packets count.

The final part of the simulation is describing a scenario where the different sources reduce their packet-sizes. This occurs in the interval from 32 seconds to 43 seconds. First out is the BE flow to reduce its packet-sizes from 80 bytes to 50 bytes. The impact of this alteration on the BE flow itself is naturally an increase in the received packets count since the packets are smaller, but also an increase in packets lost since the amount of packets lost will increase when using smaller packets, although the amount of lost bits does not increase. For both the AF and the EF flow the decreased BE packet-sizes are irrelevant, the alteration has no influence on the performance of either of the flows.



(a) EF-flow

(b) Throughput

Figure 6.5: **Received packets at the receiver, loss-rate and packets lost during transmission for the EF classified flow. At the right the total system throughput is plotted.**

The BE packet-sizes are increased to their initial value of 80 bytes after 35 seconds, before the AF source decreases its packet-sizes from 80 bytes to 50 bytes after 36 seconds. These packet-sizes are used in the following 3 seconds before the packet-sizes are increased to their initial value after 39 seconds. The impact of the AF packet-size reduction on the BE flow is a decrease in the received packet count and a corresponding increase in the lost packet count. This is due to the increased rate of packets received by the AF flow when reducing its packet-sizes. The impact on the EF flow is none.

After 40 seconds the EF flow decreases its packet-sizes from 80 bytes to 50 bytes. The impact of this alteration on the BE flow is a reduction in the received packet count and a corresponding increase in the lost packet count. The mentioned alteration has no influence on the AF flow, while the EF flow itself does experience an increase in the received packet count since the packets are smaller.

When considering the throughput of each of the flows in Figure 6.5b it becomes evident that the reduction of packet-sizes potentially yields an increase in throughput. For the reduction of packet-sizes for each of the different flows the total system throughput increases to its theoretical limit set by the SE output bandwidth. The BE flow gets only a small increase in throughput due to the prioritizing of the flows, while the two other flows fulfills their individual potential for increase in throughput when reducing their packet-size. Since the resources are sparse, one of the flows must experience a degradation in performance when another flow gets to increase its performance. The loosing flow is as always the lowest prioritized BE flow, resources are deprived from it for the good of the throughput performance of the two other flows.

6.3 Further Readings

In the following section a collection of literature where DRR can be studied further is given. Further description of DWRR and a comparison to the Worst-case Fair Weighted Queueing+ algorithm is given in [45]. In [46] the advantages of using DWRR in multiple access interference (MAI) control in CDMA networks is shown. The use of DWRR in Physical Frame Time-slot Switching is proposed in [47].

6.4 Preliminary Conclusions

DWRR inherits the fairness and low complexity advantages of conventional DRR while it also introduces the capability of differentiation between different flows. When considering delay, DWRR exhibits much of the same characteristics as of SP. The highest weighted flow (EF) performs more or less independent of the behavior of the other flows. In contrast, the lowest weighted flow (BE) pays the price of the performance of the higher weighted flows. When considering throughput, the statistics are similar; the two highest weighted flows performs independent without flaw while the lowest weighted flow is very dependent of the behavior of the other flows and of the SE output bandwidth. Although the characteristics exhibited by DWRR in the simulations are very similar to the unfair SP, the possibility of tuning the weights of each flow to any arbitrary combination makes DWRR both very dynamic and potentially as fair as conventional DRR. The mentioned features together with the low complexity makes the DWRR a very interesting algorithm when considering high-speed links handling flows with different QoS requirements.

Chapter 7

Queue Capacity Considerations

The capacities of the queues used in the different scheduling algorithms influence both the delay performances and the throughput performances of the different flows. Small queues imply short average delay, at the cost of an increased risk of overflow, while in contrast, large queues reduce the probability of packet-loss but increases average delay.

7.1 Altering Capacities

The different queue-sizes used in the simulations span from 10 kB to 240 kB. The capacity of the queues were increased for each simulation run, adding 10 kB to the capacity for each run, meaning simulating first with a capacity of 10 kB, then 20 kB, before 30 kB and so on. The different algorithms were handled separately, collecting only results from one algorithm at the time. When deploying the different algorithms the other system parameters, such as SE output bandwidth, source data-rates, etc. were set static to default values, the only altered parameters was the queue capacity parameter. The static parameters were set to the default value used throughout parts of this thesis; the three sources transmitted with equal rate, 44.8 Mbps, the SE output bandwidth was set to 175 Mbps and the different prioritizing and queue weights equal to the ones used prior in this thesis.

In order to make the simulation results of the different algorithms comparable to one another, the total queuing capacity of the SE was set equal for all algorithms for a given scenario. The FIFO approach is the only approach where only one queue is deployed inside the SE; FIFO uses all of the allocated queuing memory in a single queue. In contrast, the other algorithms (SP, DRR, DWRR) routes the different data-flows onto individual queues. This means effectively implementing three FIFO drop-tail queues inside the SE for the given scenario considered throughout the thesis where a BE flow, a AF flow and a EF flow are classified individually.

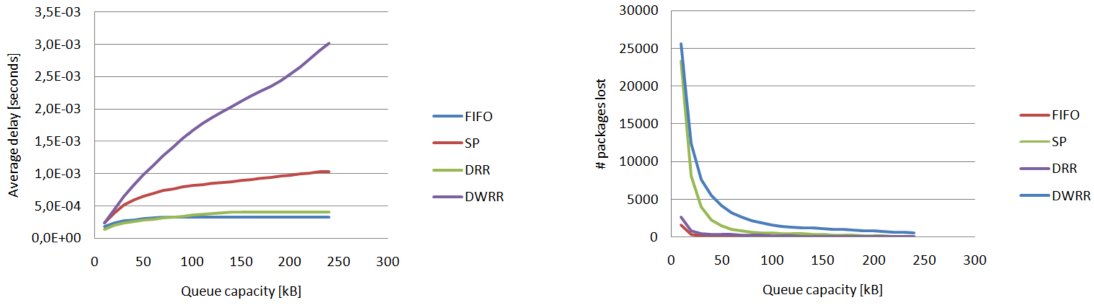
When using three different internal queues in the SE, the allocated queuing memory is divided in three. In an example, if the FIFO approach is allocated 30 kB of memory in a given scenario, all of the allocated memory is used in a single queue, while the corresponding scenario for one of the three other algorithms, is three internal FIFO drop-tail queues which each have a capacity of 10 kB.

7.2 Delay and Packet-loss

The delay of a given flow is naturally dependent of the size of the queue that is handling the flow. Small queue yields low delay with high probability of packet-loss, large queues the opposite. When considering bursty flows, where many packets arrive inside a small time-interval, before potentially no packets arrive at all in the following time-interval, a large queue would work as a buffer in such a scenario, smoothing the burstyness out, always having a packet at the front of the queue ready for transmission and never experiencing overflow. When considering the same scenario for a small queue, the delay would be reduced as only recently arrived packets are allowed onto the queue, older packets are lost due to the higher probability of overflow.

The impact of queue-sizes is specially significant when considering scheduling algorithms which prioritize between different flows. In example, if a queue handling a high prioritized flow is large, it is as mentioned more likely that a packet is present at the front of the queue, ready for transmission, at any given time, than what is the case for a small queue. And when prioritized high, a flow is more likely to be allowed transmission of a packet instead of a lower prioritized flow when both have packets ready for transmission. This means that larger queues would potentially increase the performance of higher prioritized flows, while lower prioritized flows could experience a decrease in performance.

In Figure 7.1a the delay characteristics for the BE flow when using different queue capacities are shown. Starting with FIFO, the average delay experienced when using the FIFO algorithm is slightly increasing as the capacity increase from 10 kB to approximately 75 kB, before it stabilizes at a more or less constant delay for larger queue capacities. When compared to the other approaches, the FIFO algorithm offers the best performance of all from approximately 100 kB of allocated memory for queuing, while DRR is the better one from 10 kB to approximately 100 kB of allocated memory. The delay when using SP is longer than for the two mentioned algorithms, increasing rapidly for allocated memory between 10 kB and approximately 35 kB, before a smaller increase is experienced for the remaining capacities. The worst performing approach for any capacity is DWRR. The delay increases rapidly as the capacity increases for all capacities. This is due to that an increase in queue capacity naturally yields more room for packets to be stored, and as more packets can be stored in a queue, the average age of the packets increase, leaving an increased average delay and a potential decrease in packet-loss. The delay statistics should be considered together with the packet-loss statistics of the different flows. The packet-loss statistics of the BE flow is shown in Figure 7.1b. When comparing Figure 7.1a with Figure 7.1b, note that the colors of the lines representing the statistics of the different algorithms are different from one figure to the other. As for the delay statistics, the two algorithms which offers similar and best performances for the BE flow are DRR and FIFO when considering queue capacity requirements. The number of packets lost rapidly decreases toward zero when increasing the queue capacity to approximately 35 kB when deploying these algorithms. As for SP, the number of packets lost decrease from approximately 22500 when allocating 10 kB memory for queuing toward zero for 100 kB of allocated memory. Since the BE flow is the least prioritized flow, the SP only uses resources on it when the two higher prioritized



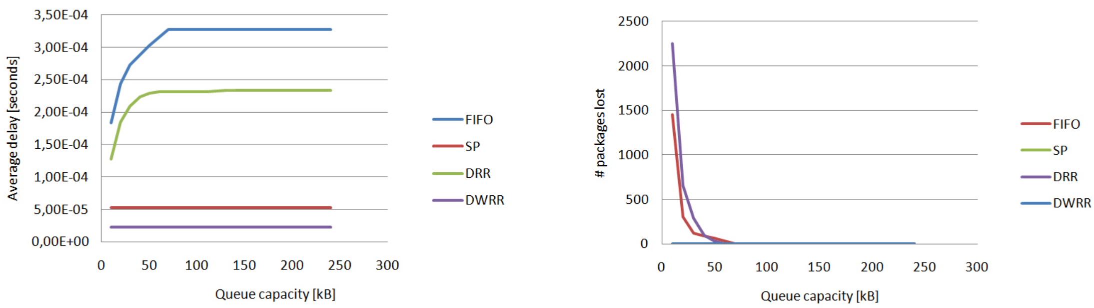
(a) Delay

(b) Packet-loss

Figure 7.1: **Delay and packet-loss statistics of the BE classified data-flow.**

flows performs close to perfection. The DWRR algorithms offers, similar to the delay statistics, the worst packet-loss statistics for the BE flow of the implemented algorithms. The packet-loss is above 25000 when allocating 10 kB of memory to queuing, and does after a rapid decrease in packet-loss toward 100 kB, only exhibit a slight decrease in packet-loss for the remaining capacities, never reaching zero. The high delay observed in Figure 7.1a and the slow decrease in packet-loss 7.1b makes the DWRR algorithm the one algorithm offering least performance for the BE flow when considering queue capacity. This is though relative, since the weights used in DWRR could be altered, leaving different performance statistics.

When considering Figure 7.1a, it seems that for low queue capacities there is only a marginal difference in performance for the different algorithms. A more complete description of the performances of the different algorithms is though observable when considering Figure 7.1a together with Figure 7.1b. As for the delay statistics only exhibiting small differences when using small queues, the packet-loss statistics exhibits extreme differences when considering small queues. In the other end of the capacity scale, the opposite characteristics can, as expected, be observed; large delay and little packet-loss with large queues.



(a) Delay

(b) Packet-loss

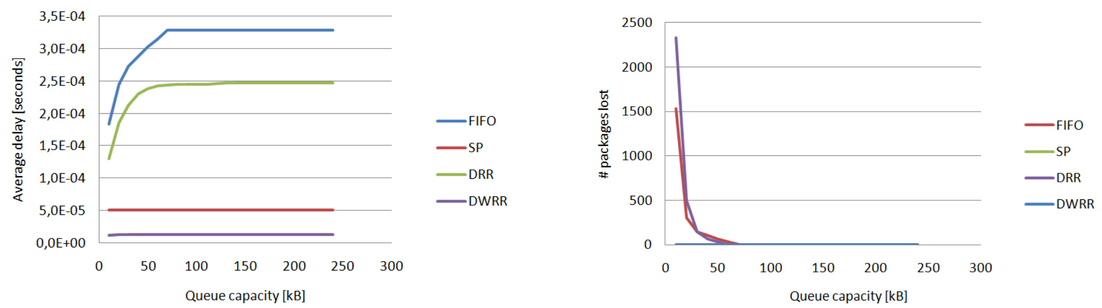
Figure 7.2: **Delay and packet-loss statistics of the AF classified data-flow.**

The delay statistics when altering the queue capacities of the AF flow is shown in Figure 7.3a.

The statistics are very different from the statistics of the BE flow, for the AF flow it is the DWRR algorithm that offers the best performance when considering queue capacity. This is in contrast to the statistics of the BE flow, where DWRR offered the poorest performance. As can be observed, when using DWRR or SP, the delay stay constant for all queue capacities, where DWRR operates with half the delay of SP. DRR is the third best algorithm, exhibiting an increase in delay for queue capacities between 10 kB and approximately 75 kB, before staying constant for the remaining capacities. FIFO offers the longest delay of the different algorithms, requiring approximately 100 kB of memory allocated to queuing before a constant value of delay is reached.

Similar as for the delay statistics, it is DWRR and SP which offers least packet-loss of the different algorithms. The packet-loss of the two mentioned algorithms is zero for all queue capacities. DRR and FIFO exhibits significant packet-loss for low capacities, FIFO requiring approximately 75 kB of memory for zero packet-loss for the AF flow, while DRR requires approximately 50 kB. The lower threshold of the DRR does though not tell the whole story, DRR exhibits larger packet-loss than FIFO for capacities between 10 kB and approximately 45 kB.

The contrast in delay statistics and packet-loss statistics between the AF flow and the BE flow is significant. FIFO and DRR offers best performances for the BE flow due to the fairness of the algorithms, resources are allocated equally between the different flows. DWRR and SP offers best performances for the AF flow since they effectively and directly prioritize between the different flows, offering higher performances for higher prioritized flows (AF) at the price of poorer performances of the lower prioritized flows (BE). The same arguments explain the differences in packet-loss statistics between the BE flow and the AF flow. In Figure 7.3a the



(a) Delay

(b) Packet-loss

Figure 7.3: **Delay and packet-loss statistics of the EF classified data-flow.**

delay statistics of the EF flow is shown. The main characteristics are very similar to the ones of the AF flow. The only difference is a slightly smaller constant delay when using DWRR. As for the packet-loss statistics, observable in Figure 7.3b, the differences between the AF flow and the EF flow are not possible to distinguish.

When comparing the statistics of the BE flow with the EF flow, the differences are almost identical to the differences between the BE flow and the AF flow. The analysis of the differences between the BE flow and the AF flow described above also holds for the differences between the

BE flow and the EF flow, the differences occur of the same reasons for both comparisons. Using the mentioned analysis above and remembering that the EF flow is the highest prioritized flow, the delay statistics and packet-loss statistics of the EF flow are as expected.

7.3 Comments

Both the delay and packet-loss statistics of the different flows are naturally dependent of the capacities of the different queues. When deploying the differentiating algorithms (SP, DWRR) the different classified flows experience performances corresponding to the prioritizing of the flows. The capacities of the different queues are to be considered as any other transmission resource, and thus the performances of lowest prioritized flows are degraded at the fortune of stable and good performances of the higher prioritized flows. Of the different algorithms compared to one another in the section above, it seems that SP is the one approach offering the highest promises. The low delay and lack of packet-loss of the highest prioritized flows together with the lower delay and packet-loss cost when considering the lowest prioritized flow makes SP the most memory effective algorithm with differentiation capabilities. It must though be mentioned that this is in a scenario where only queue capacity resources are sparse. Also, the dynamic capabilities of DWRR, where altering flow weights alters the performance statistics, makes it hard to make any bombastic conclusion. The fact that the price of memory is very low in the recent years makes the cost of queue capacity only secondary when designing a hardware entity. An interesting feature of the queue capacity consideration is though, that one should not increase the capacities of the queues above certain thresholds. When loss-less transmission for all flows is reached, a further expansion of queue capacity only introduces more delay than needed for lower prioritized flows. Some overhead should though be available as the data-rates seldom are constant, a burst in data-rate should not result in massive packet-loss.

Chapter 8

Impact of Fading

Signal strength is a major factor in any transmission budget. Variation in signal strength will influence the performance of wireless transmission and thus it is a major concern for providers of services utilizing wireless communication. Deployment of smarter scheduling algorithms in nodes which experience fading in signal strength could potentially offer network providers more fading-resistant solutions where the capability of differentiation between different classified data-flows enhances the overall performance. Differentiation between data-flows is, as mentioned in Section 2.1, a valuable feature since different types of data-flows have different requirements concerning delay, loss, etc.

8.1 Realization of Fading

In order to simulate the characteristics of fading; loss of signal strength, a simplification were used in the simulations. Instead of simulating a complete scenario with modulation, path loss etc, the effective reduction of available resources were realized through simply reducing the bandwidth of the SE output. The different scheduling algorithms deployed in this thesis have low complexity for use in high speed links, the focus is on differentiation/fairness between several flows over a shared link. Since all data-flows are using a shared link from the SE to their respective receivers and the focus is as mentioned, the described simplifications in the simulations could be made.

The mentioned realization, where the SE output bandwidth were to be altered for simulating fading characteristics, was done as follows. The simulation setup was partially the same as for the simulation scenarios in the previous chapters, where three separate sources were classified as either BE, AF or EF. The EF flow is the highest prioritized flow, while the AF flow is the second prioritized flow, and the BE flow the least prioritized flow. Each of the sources transmits with a data-rate of 60 Mbps. The total queue capacity for the SE is identical for all algorithms, 60 kB. Note that for the conventional FIFO algorithm with only one internal queue in the SE, the whole capacity of 60 kB is made available for the single queue. For the other algorithms, where each of the different flows are routed onto individual queues, the available 60 kB are divided into three 20 kB capacity FIFO drop-tail queues.

The main parameter in these simulations, the SE output bandwidth, was altered for multiple simulation runs. The bandwidth were altered from 10 Mbps to 200 Mbps in steps of 5 Mbps. Each of the simulation runs yielded individual system performance characteristics. The average delay, the loss-rate and the average throughput were collected for each run. In the following sections the collected results are presented for each flow using different algorithms.

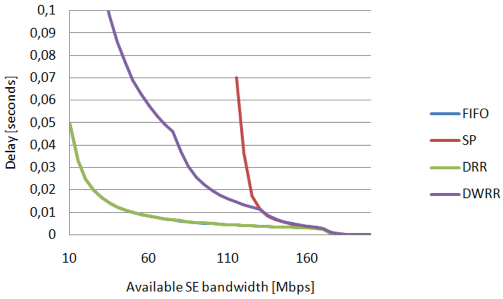
8.2 Delay

In chapters 3, 4, 5 and 6 the simulations of each of the algorithms exhibit a dependence between the available SE output bandwidth and the delay experienced by each flow. The following section presents a more detailed description of the mentioned dependence in a more systematic manner.

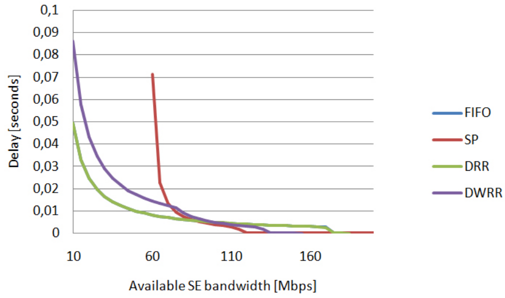
In Figure 8.1a the different delay characteristics experienced by the BE classified flow for different amounts of available bandwidth is shown. Note that the FIFO distribution is more or less identical to the DRR distribution as the packet-sizes are constant and equal for all sources. Due to the prioritizing of the flows in SP queuing and DWRR queuing, where the BE flow is least prioritized, the BE flow experience best conditions when deploying the algorithms which does not differ between the different flows, namely FIFO and DRR. As for the two other algorithms, the DWRR algorithm offers better performances for the BE flow than the SP algorithm. Since the weighted deficit amount of bits calculated in DWRR always are non-zero, there will always be some capacity reserved for the least prioritized flow. The effective amount of bits reserved is strictly dependent of the weights designated to each flow.

For the SP approach the story is quite different, the BE flow is not allowed to transmit anything before both the EF flow and the AF flow performs close to perfection. This means that all packets are lost when the available recourses are lesser than a given threshold. This can be observed in Figure 8.1a, where the SP delay distribution goes toward infinity for available SE bandwidth smaller than approximately 110 Mbps. When the two higher prioritized flows perform perfectly, all the abundant bandwidth is made available for the BE flow, yielding a sharp increase in delay performance when first allocated any resources at all.
The BE flow does not perform without a quite significant delay until the amount of available bandwidth reaches approximately 175 Mbps for any of the algorithms. This either due to equally distributed resources between the different flows (FIFO, DRR), or because the BE flow is least prioritized when deploying differentiating algorithms (SP, DWRR). When differentiating between the flows, the two other flows must experience close to perfect conditions before the BE flow gets resources.

In Figure 8.2b the BE delay distribution for near perfect performance is shown. It is evident that all of the algorithms perform very similar with respect to delay when the amount of available bandwidth reaches approximately 180 Mbps. The only differences are that the DRR approach and the FIFO approach reaches lower delay for slightly smaller SE output bandwidth than the SP approach and the DWRR approach. When the amount of resources reaches approximately 185 Mbps, the delay exhibit some stabilization. In this scenario the DRR approach performs



(a) BE-flow

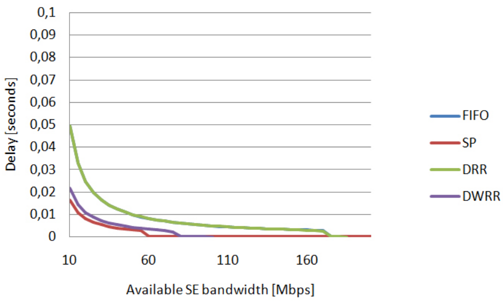


(b) AF-flow

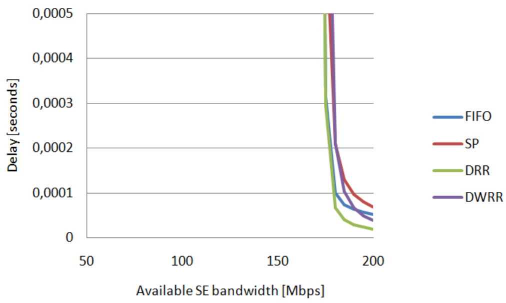
Figure 8.1: **Delay statistics of the BE classified data-flow and of the AF classified data-flow.**

best, DWRR second best, FIFO third best and the SP approach yields highest delay for higher amounts of available SE output bandwidth. Note that the performance for scenarios where the available bandwidth is above 200 Mbps are not targeted for further discussion.

The delay experienced by the AF flow for different amounts of SE output bandwidth is shown in Figure 8.1b. The delay distribution for the FIFO approach and the DRR approach are both identical to each others and to their respective distributions for the BE flow. This is again due to the lack of differentiating between the different flows, all the flows compete equally and are thus distributed an equal amount of the bandwidth. The FIFO approach and the DRR approach does, as mentioned previously for the BE flow, need the SE output bandwidth to be at least approximately 180 Mbps to operate as desired.



(a) EF-flow



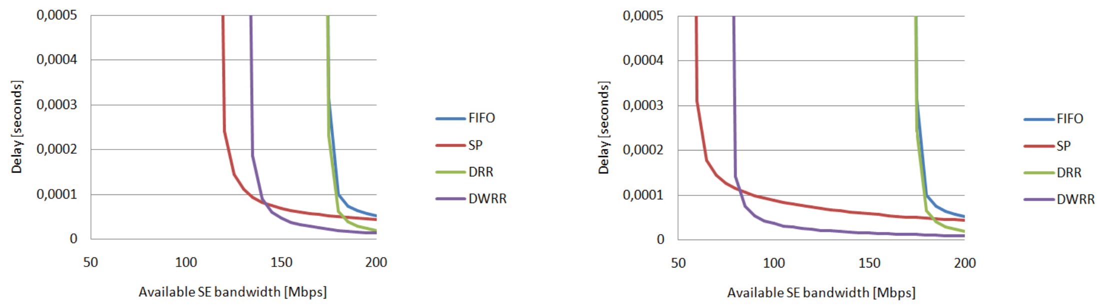
(b) BE-flow

Figure 8.2: **Delay statistics of the EF classified data-flow and a more detailed overview of the BE classified data-flow.**

The SP approach yields better performances for the AF flow when less bandwidth is available when compared to the BE flow, approximately from 115 Mbps and upwards of available bandwidth is needed for the AF flow to operate with low delay. Below 60 Mbps of available bandwidth the delay goes to infinity, as all resources are used to secure the performance of the higher prioritized EF flow.

When deploying the DWRR algorithm, the AF flow needs approximately a SE output bandwidth of 125 Mbps to operate without much delay. This means that the DWRR approach needs more bandwidth than the SP approach for the AF flow to operate flawlessly. This is because some resources always are reserved for the BE flow when using DWRR, while for SP all resources above the threshold set by the performance of the EF flow are distributed to the AF flow. The DWRR delay distribution is of course dependent of the weights of each respective flow, different weighting of the flows would yield different delay distributions.

Figure 8.3a shows the delay distribution of each of the approaches when the delay is low enough for near perfect performance for the AF flow. The FIFO algorithm is the one algorithm needing the most bandwidth, approximately 180 Mbps, for proper AF performance. This is, as mentioned, due to the equal sharing of resources in the FIFO approach. The DRR approach offers almost identical delay characteristics as the FIFO approach, the only difference is the slightly lower delay when considering scenarios where the available bandwidth is larger than approximately 185 Mbps. Furthermore, the SP approach offers the lowest requirement of available bandwidth for delay less than 0.0005 seconds. From approximately 140 Mbps the SP approach offers AF performance with very little delay. The DWRR approach needs slightly more bandwidth than the SP approach for the AF flow to perform with delay between approximately 0.0001 seconds and 0.0005 seconds. For delay less than approximately 0.0001 seconds, the DWRR approach needs less available bandwidth than the SP approach. For the AF flow, the DWRR algorithm is the one algorithm which offers lowest delay when considering 200 Mbps to be the maximal amount of available bandwidth.



(a) AF-flow (b) EF-flow

Figure 8.3: **More detailed delay statistics of the AF classified data-flow and of the EF classified data-flow.**

Finally, the delay characteristics of the EF flow is shown in Figure 8.2a. The DRR approach and the FIFO approach have delay distributions which are not possible to distinguish from one another in the mentioned figure, in other words, they perform more or less identically. The DWRR approach requires approximately 75 Mbps of available bandwidth for transmission with very little delay, approximately 100 Mbps less than the requirements of both the DRR approach and the FIFO approach. Again, the SP algorithm offers the lowest requirement for good performances when considering delay, only 60 Mbps is needed for the EF flow to perform with very little delay when setting the EF flow as the highest prioritized flow.

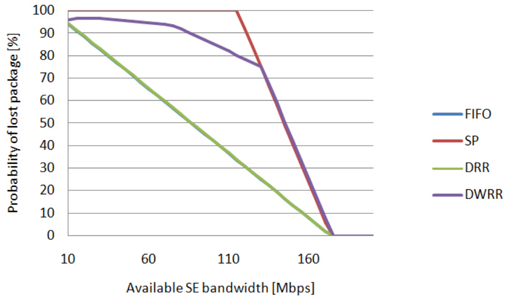
In Figure 8.3b the delay distributions for the EF flow when the delay is less than 0.0005 seconds is shown. FIFO needs approximately 180 Mbps for delay less than 0.0001 seconds, while the DRR approach has a delay of approximately 0.000075 seconds with the same approximately 180 Mbps of available bandwidth. DWRR performs better, only approximately 80 Mbps is needed for delay less than 0.0001 seconds. SP require the least amount of available SE output bandwidth for performing with delay between 0.0002 seconds to 0.0005 seconds, while for delay less than approximately 0.0001 seconds, the DWRR approach requires the least amount of bandwidth.

A characteristic which is shared by all the algorithms is that the derivative of the delay distributions for delays less than approximately 0.0001 seconds is very small. This means that reducing the delay further than 0.0001 seconds is very expensive when considering bandwidth.

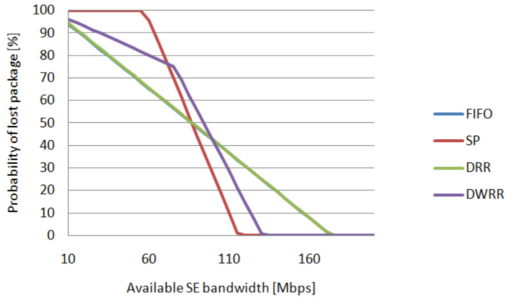
8.3 Loss-rate

Similar to the dependence between the delay experienced by each flow and the available bandwidth, there is a dependence between the loss rate experienced by each flow and the available SE output bandwidth.

The loss-rate distributions of the BE flow for different amounts of available bandwidths are shown in Figure 8.4a. The loss-rate distributions when applying FIFO and DRR can not be distinguished from one another in the figure, their loss-rate characteristics are more or less identical. The loss-rate of FIFO and DRR does not reach a level less than 10% until there is approximately 160 Mbps of available bandwidth. SP and DWRR operates with a loss-rate of approximately 30% with the same 160 Mbps of available bandwidth. Similar as for the delay considerations in the previous section, the FIFO and DRR approaches offers best performance for the BE flow, since all resources are distributed equally between all three flow, in contrast to the prioritized resource distributions of SP and DWRR. The nature of the SP approach dictates that the two higher prioritized flows must perform with 0% loss-rate before the BE flow is allocated any resources, yielding no transmission at all for the BE flow before the available bandwidth reaches approximately 130 Mbps. Although DWRR offers loss-rates less than 100% for all amounts of available bandwidth, this feature does not give DWRR any advantage over SP when applying the current flow weights, since the loss-rate distributions of SP and DWRR are identical for loss-rates less than approximately 75%. Loss-rates of the magnitude of 75% is of cource of no practical interest, but DWRR does in principle exhibit more fairness than SP.
In Figure 8.5b the loss-rate distributions of the BE flow for loss-rates less than 5% is presented. Again FIFO and DRR perform identically, needing approximately 180 Mbps of available bandwidth for operating without loss. The same amount of bandwidth is needed by the two other approaches, SP and DWRR, but their loss-rate is higher for lower amounts of available bandwidths than for FIFO and DRR. DWRR performs worst, the possible attractive feature of never reaching 100% loss-rate for the BE flow is of little interest when it needs the most amount of bandwidth of all algorithms for performance with less than 1% loss. As mentioned before for DWRR, it is though impossible to judge DWRR finally as the flow weights could easily be altered, leaving different performance characteristics.

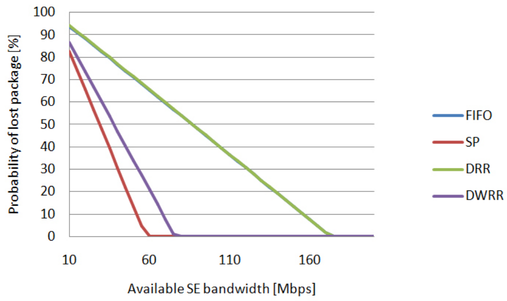


(a) BE-flow

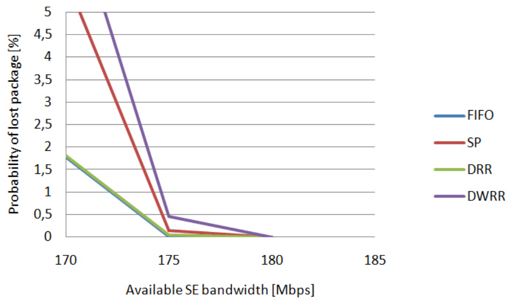


(b) AF-flow

Figure 8.4: **Packet-loss statistics for the BE classified data-flow and of the AF classified data-flow.**



(a) EF-flow

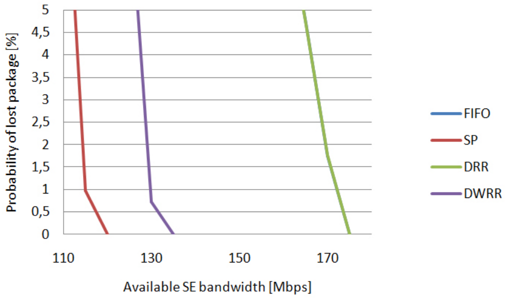


(b) BE-flow

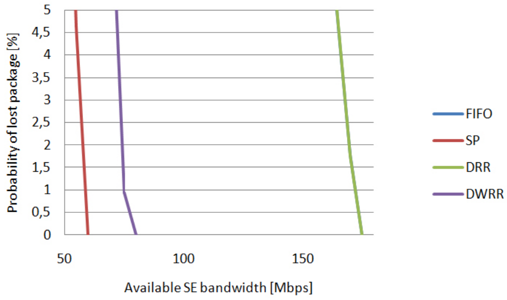
Figure 8.5: **Packet loss statistics for the EF classified data-flow and packet loss statistics with increased detail of the BE classified data-flow.**

The AF flows loss-rate distributions is shown in Figure 8.4b. FIFO and DRR does again perform very similarly. As for the BE flow, the AF flow require approximately 180 Mbps for performing without loss when applying FIFO and DRR. SP has a loss-rate of 100% until the SE output bandwidth reaches at least approximately 70 Mbps, where it starts a rapid decline in loss-rate per added bandwidth. DWRR does again offer some throughput for all amounts of available bandwidths, but since the loss-rate is in the magnitude of 75% when the SP approach passes the DWRR approach in loss-rate per bandwidth, the SP approach reaches loss-rates where transmission of data could possibly be realized at lower amounts of SE output bandwidth than the DWRR approach. SP reaches loss-rates in the magnitude of 1% at approximately 115 Mbps, while DWRR require at least approximately 130 Mbps for the same performance.

These low loss-rate characteristics can be observed in more detail in Figure 8.6a, where both FIFO and DRR reaches zero loss at approximately 175 Mbps, while DWRR requires approximately 135 Mbps of SE output bandwidth for zero loss performance and SP approximately 120 Mbps. As for the BE flow, is must be mentioned that a final conclusion to the performances of the DWRR approach is hard to find as alterations of the weights assigned to each flow could be



(a) AF-flow



(b) EF-flow

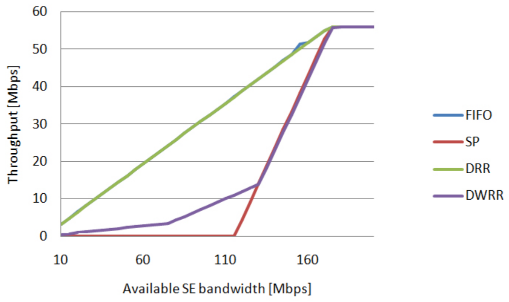
Figure 8.6: **Packet-loss statistics with increased detail for the AF classified data-flow and of the EF classified data-flow.**

altered, offering different performances.

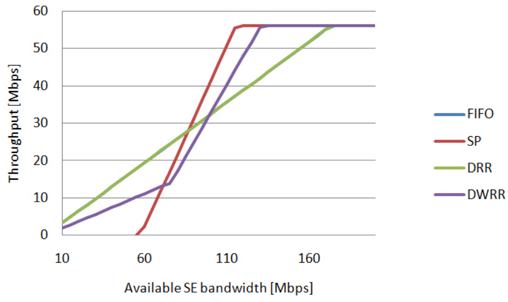
The EF flow loss-rates is plotted in Figure 8.5a. The EF flow is as mentioned the highest prioritized flow when applying SP and DWRR, while there is no differentiation between the flows in FIFO or DRR. FIFO and DRR perform identically, both require approximately 175 Mbps bandwidth for loss-rates in the magnitude of 1 %. DWRR requires approximately 80 Mbps for the same performance, while SP only requires approximately 60 Mbps. The low requirement of the SP is due to that all available resources are allocated to the EF flow when performing less than perfect. The low loss-rate characteristics can be studied in greater detail in Figure 8.6b.

8.4 Throughput

The throughput of each of the flows when applying the different algorithms can be observed in figures 8.7a, 8.7b and 8.8. Taking into account the results from the delay distributions and the loss-rate distributions mentioned above, it is hardly a surprise that the BE flow (Fig. 8.7a) requires at least a SE output bandwidth of approximately 180 Mbps to reach a throughput to the receiver identical to the data-rate of the source. The threshold of 180 Mbps holds for all of the algorithms, while both FIFO and DRR offers greater throughput per available bandwidth than SP and DWRR. Since the AF flow and the EF flow is prioritized higher than the BE flow, SP offers no throughput at all for the BE flow before the amount of available bandwidth reaches levels where both the AF flow and the EF flow performs close to perfection. In contrast to the SP approach, the DWRR approach offers more fairness, the throughput is always above none-zero, although the throughput distributions of SP and DWRR are identical from a throughput of approximately 25 Mbps and upward.
The throughput distributions of the different algorithms for the AF flow is shown in figure 8.7b. FIFO and DRR exhibits identical throughput distributions, both requiring a SE output bandwidth of 180 Mbps for operating without congestion. The SP approach requires only approximately 100 Mbps of available bandwidth for the AF throughput to reach the same level as the AF source data-rate. DWRR is the second most effective algorithm with respect to



(a) BE-flow



(b) AF-flow

Figure 8.7: **Throughput for the AF classified data-flow and for the EF classified data-flow.**

throughput per bandwidth, requiring approximately 115 Mbps of SE output bandwidth for reaching its maximum throughput rate.

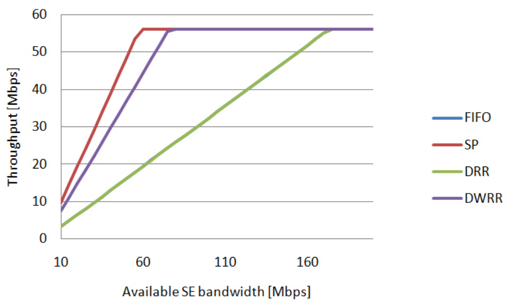


Figure 8.8: **Throughput of the EF classified data-flow versus available SE bandwidth.**

For the differentiating algorithms, the EF flow is the highest prioritized flow. The impact of this prioritizing on the throughput of the flow can, together with the throughput characteristics of the non-differentiating algorithms, be observed in Figure 8.8. The distributions of the non-differentiating algorithms, FIFO and DRR, are also identical for the EF flow, slowly increasing from a throughput of 5 Mbps for a SE output bandwidth of 10 Mbps, toward a maximum throughput of 60 Mbps for approximately 180 Mbps of available SE output bandwidth. DWRR reaches the maximum throughput threshold with less bandwidth available, only approximately 70 Mbps of bandwidth is required for maximum EF flow throughput. The SP approach is the most effective approach for the EF flow, reaching the maximum throughput threshold with only a approximately 60 Mbps SE output bandwidth available.

8.5 Comments

The impact of fading is significant in any wireless transmission system. The different algorithms described in this thesis handles fading differently. FIFO and DRR distributes the available resources equally among the different flows, while SP and DWRR differentiate between the different flows. When considering FIFO and DRR, the impact of fading is naturally equal for all flows, they experience equal delay, probability of packet-loss and throughput statistics. To reach flawless transmission for any of the queues, the available bandwidth must at least be equal to the sum of the data-rates of the different flows. For SP the statistics are quite different. Due to the strict prioritizing all available resources are allocated to the highest prioritized flow, and furthermore the highest prioritized flow must operate flawlessly before any resources are allocated to the second prioritized flow. The same counts for the third prioritized flow; the second prioritized flow must operate perfectly before the third prioritized flow is allocated any resources. Since DWRR exhibits more fairness, it requires a larger amount of available resources for the highest prioritized flow to operate flawlessly. The highest prioritized flow does not get all the resources when resources are sparse, some are reserved both the second prioritized flow and the third prioritized flow. The capability to differentiate between the flows and its overall good performance together with the dynamic features of the alterable flow weights makes the DWRR approach the most promising approach when considering how the different algorithms handles fading.

Chapter 9

Discussion and Conclusion

Finding algorithms with low complexity offering fairness among and potentially differentiation between different data-flows is important in the evolution of communication technology. The growing demand of network nodes capable of taking into account the different QoS requirements of different flows to better utilize the available resources at the same time as some degree of fairness is maintained, makes more intelligent packet scheduling a central topic in future development of communication technologies.

9.1 Non-differentiating Algorithms

Two algorithms without capabilities to differentiate between different data-flows are discussed in this thesis:

- First-In-First-Out in Chapter 3
- Deficit Round-Robin in Chapter 5

First-In-First-Out (FIFO) is the least complex algorithm mentioned in this thesis. It has no differentiation capabilities and offers no guarantees with respect to fairness. The simple implementation has made it the standard approach to scheduling in conventional network nodes. Communication technology has evolved rapidly over the last decade and more intelligent algorithms for scheduling packets which are capable of both exploiting different transmission scenarios and differentiating between data-flows with different QoS requirements is needed in future networks. FIFO is most suitable in scenarios where the different data-flows handled exhibits coherent transmission statistics with respect to packet-size and data-rate. As seen in the simulations in Chapter 3, FIFO lacks capabilities of differentiation between flows and exhibits potentially unfair handling. The mentioned characteristics makes FIFO unsuitable for modern high-speed point-to-point networks handling multiple data-flows, each with potentially different traffic characteristics and QoS requirements.

Deficit Round-Robin (DRR) is another algorithm without capabilities of differentiating between different flows. DRR distributes the available resources fairly among the different flows in an iterative manner. The deficit counter feature makes DRR robust against ill behaving sources, as seen in the simulations in Chapter 5, the throughput of one data-flow is not dependent of the characteristics of the other flows. Only a slight difference in average delay for one flow can be observed as a result of alterations in the traffic characteristics of other flows. The lack of capabilities to differentiate between different data-flows is a major drawback to DRR. In order to schedule packets in an effective manner with respect to the different services offered in modern communication technology, a differentiation in allocation of resources between different data-flows is required. Although the capability of differentiation lacks, the attractive fairness properties of DRR in combination with its low complexity still makes it an interesting algorithm for packet scheduling, favorably in networks where QoS requirements of the different flows are similar, but where packet-sizes and data-rates are varying.

9.2 Differentiating Algorithms

The two algorithms with capabilities to differentiate between different data-flows are discussed in:

- Strict Priority in Chapter 4
- Deficit Weighted Round-Robin in Chapter 6

Strict Priority Queuing (SP) does as mentioned offer differentiation between different data-flows. The data-flows are classified and then strictly prioritized with respect to the classification when scheduling. As seen in Chapter 4, the strictness of the algorithm goes as far as allocating all resources for the highest prioritized flow until the available resources reaches such an amount that the highest prioritized flow operates without flaw. Not until such a performance for the highest prioritized flow is reached, are any resources allocated to any lower prioritized flow. The performances of the lower prioritized flows are totally dependent of the characteristics of the higher prioritized flows, a property that is potentially not very attractive in certain communication systems. Although SP offers capabilities to differentiate between flows with respect to different QoS requirements, the obvious potentially very unfair properties of SP makes it not suitable as a standalone algorithm in conventional networks transporting services for public use. SP is more suitable in specialized networks or in use in hierarchical structure of scheduling algorithms, as briefly mentioned in Section 2.6.

Deficit Weighted Round Robin (DWRR) inherits the low complexity of DRR together with a potential for equally fair scheduling as of DRR. Based upon the features DRR, DWRR also includes individual weighting of the different data-flows target for scheduling. By simply weighting the deficit counters used for deciding which flow is to be serviced, DWRR offers a dynamic differentiation solution. The flows can be assigned any combination of weights, potentially yielding performances effectively spanning from the unfair characteristics of SP to the fair characteristics of DRR. The low complexity, potential fairness and capability of differentiating between flows exhibiting different QoS requirements, makes DWRR the most promising algorithm discussed in detail in this thesis. DWRR has the largest potential when considering a high-speed point-to-point scenario where different data-flows, representing a multiple of services, are transported over one link.

Bibliography

Bibliography

[1] "Weighted Fair Queueing and Compensation Techniques for Wireless Packet Switched Networks", D-S.Lee, C-M.Chen, C-Y.Tang, IEEE Transactions on Vehicular Technology, Vol. 56, No 1, January 2007

[2] "Wireless Communications", A.Goldsmith, ISBN 0-521-83716-3

[3] "Computer Networks Fourth Edition", A.S.Tanenbaum, ISBN 0-13-038488-7

[4] "Efficient Fair Queuing Using Deficit Round-Robin", M.Shreedhar, G.Varghese, IEEE/ACM Transactions on Networking, Vol. 4, No 3, June 1996

[5] "Asymptotic Analysis of Proportional Fair Algorithm", J.M.Holtzman, IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Vol. 2, October 2001

[6] "MAC 20-5 - Proportional Fair Scheduling: Analytical Insight under Rayleigh Fading Environment", L.Erwu, K.K.Leung, IEEE Wireless Communications and Networking Conference, April 2008

[7] "Opportunistic Beamforming using Dumb Antennas", P.Viswanath, D.N.C.Tse, R.Laroia, IEEE Transactions on Information Theory, Vol 48, Issue 6, June 2002

[8] "Joint an Advanced Proportionally Fair Scheduling and Rate Adaptation for Multi-services in TDD-CDMA Systems", P.Mueng, W.Yichuan, W.Wenbo, IEEE 59th Vehicular Technology Conference, Vol 3, May 2004

[9] "A Proportionally Fair Scheduling Algorithm with QoS and Priority in 1xEV-DO", K.Kuenyoung, K.Hoon, H.Youngnam, IEEE Symposium on Personal, Indoor and Mobile Radio Communications, Vol 5, Septemer 2002

[10] "Spectral Efficiency and Fairness for Opportunistic Round Robin Scheduling", V.Hassel, M.Røed Hanssen, G.E.Øien, IEEE International Conference on Communications, Vol. 2, June 2006

[11] "High Performance Service-time-stamp Computation for WFQ IP Packet Scheduling", C.McKillen, S.Sezer, X.Yang, IEEE Emerging VLSI Technologies and Architectures, June 2006

[12] "Efficient Fair Queuing Algorithms for Packet-Switched Networks", D.Stiliadis, A.Varma, IEEE Transactions on Networking, Vol. 6, No. 2, April 1998

[13] "The Role of Packet-dropping Mechanism in QoS Differentiation", G.Quadros, A.Alves, E.Monteiro, F.Boavida, IEEE International Conference on Networks, 2000

[14] "Low-delay Dynamic QoS Scheduling for Assured Service in Diffserv Network", T.Minagawa, T.Kitami, T.Ikegami, IEEE 9th International Conference on Advanced Communication Technology, Vol. 1, February 2007

[15] "Huffman Fair Queueing: A Scheduling Algorithms Providing Smooth Output Traffic", M-T.Choy, T.T.Lee, IEEE International Conference on Communications, Vol. 2, June 2006

[16] "Hierarchical Packet Fair Queuing Algorithms", J.C.R.Bennett, H.Zhang, IEEE /ACM Transactions on Networking, Vol. 5, No. 5, 1997

[17] "A Fair Queuing Algorithm for Multiple-Streams Delay-Bounded Services", M.Liu, L-C.Wuu, L-Lin, C.Y.Tsai, IEEE International Conference on Networks, 1999

[18] "Packet Scheduling over a Shared Wireless Link for Heterogeneous Classes of Traffic", O-S.Shin, K.B.Lee, IEEE International Conference on Communications, Vol. 1, June 2004

[19] "Loss-Tolerant QoS using Firm Constrains in Guaranteed Rate Networks", A.Koubaa, Y-Q.Song, IEEE Real-Time and Embedded Technology and Applications Symposium, 2004

[20] "Dynamic WFQ Scheduling for Real-Time Traffic in Wireless ATM Links", K.Pang, X.Lin, J.Zheng, Z.Gu, IEEE International Conference on Communication Technology, Vol. 2, 2000

[21] "A Fair Scheduling Discipline for Cellular Data Services with Location-Dependent Errors", M-H.Yao, J-L.Chen, H-C.Cheng, NAFIPS International Conference, Vol. 2, July 2001

[22] "E2WFQ: An Energy Efficient Fair Scheduling Policy for Wireless Systems", V.Raghunathan, S.Ganeriwal, C.Schurgers, M.Srivastava, Proc. ISLPED'02, August 2002

[23] "WF2Q: Worst-case Fair Weighted Queueing", J.C.R.Bennett, H.Zhang, Proceedings of INFOCOM'96, March 1996

[24] "Delay Optimized Worst Case Fair WFQ (WF2Q) Packet Scheduling", X.Fei, A.Marshall, IEEE International Conference on Communications, Vol. 2, 2002

[25] "Fair Scheduling With Tunable Latency: A Round-Robin Approach", H.M.Chaskar, U.Madhow, IEEE/ACM Transactions on Networking, Vol. 11, Issue 4, August 2004

[26] "Throughput Guarantees for Opportunistic Scheduling Algorithms: A Comparative Study", V.Hassel, G.E.Øien, D.Gesbert, 2006 International Telecommunications Symposium, September 2006

[27] "Fair Scheduling in Wireless Packet Networks", L.Songwu, V.Bharghavan, R.Srikant, IEEE/ACM Transactions on Networking, Vol. 7, No. 4, August 1999

[28] "Packet Scheduling for QoS Support in IEEE 802.16 Broadband Wireless Access Systems", K.Wongthavarawat, A.Ganz, International Journal of Communications Systems, 2003

[29] "Cross Layer Design for the IEEE 802.11 WLANs: Joint Rate Control and Packet Scheduling", Q.Xia, X.Jin, M.Hamdi, IEEE Transactions on Wireless Communications, Vol. 6, No. 7, July 2007

[30] "Throughput Guarantees for Wireless Networks with Opportunistic Scheduling", V.Hassel, G.E.Øien, D.Gesbert, IEEE Transactions on Wireless Communications, Vol. 6, Issue 12, December 2007

[31] "The art of computer systems performance analysis", R. Jain, John Wiley and Sons, 1991

[32] "Schedulability Analysis of Flows Scheduled with FIFO: Application to the Expedited Forwarding Class", S.Martin, P.Minet, 20th International Parallel and Distributed Processing Symposium, April 2006

[33] "Fundamental Trade-offs in Aggregate Packet Scheduling", Z.Duan, Z-Li.Zhang, Y.T.Hou, IEEE Transactions on Parallel and Distributed Systems, Vol. 16, Issue 12, December 2005

[34] "Enhancing Throughout over Wireless LANs using Channel State Dependent Packet Scheduling", P.Bhagwat, P.Bhattacharya, A.Krishna, S.K.Tripathi, INFOCOM'96 Fifteenth Annual Joint Conference of the IEEE Computer Societies Networking the Next Generation, Vol. 3, March 1996

[35] "A First-In-First-Out Memory for Signal Processing Applications" , N.Kanopoulos, J.Hallenbeck, IEEE Transactions on Circuits and Systems, Vol. 33, Issue 5, May 1986

[36] "A Case for Simplicity in Providing Network Quality of Service: Class-based Strict Priority Queueing", J.Schmitt, F.Zdarsky, 12th IEEE International Conference on Networks, Vol. 2, November 2004

[37] "Per-flow Guarantees under Class-based Priority Queueing", J.Schmitt, P.Hurley, M.Hollick, R.Steinmetz, IEEE Global Telecommunications Conference, Vol. 7, December 2003

[38] "Modeling the Performance of Low Latency Queueing for Emergency Telecommunications", D.M. Bevilacqua Masi, M.J.Fischer, D.A. Garbin, 2007 Winter Simulations Conference, Descember 2007

[39] "Priority Queue Schedulers with Approximate Sorting in Output-buffered Switches", J.Liebeherr, D.E.Wrege, IEEE Journal on Selected Areas in Communications, Vol. 17, Issue 6, June 1999

[40] "Modeling Priority Queueing Systems with Multi-Class Self-Similar Network Traffic", J.Xiaolong, M.Geyong, IEEE International Conference on Communications, June 2007

[41] "The Effect of Quantum Size on Deficit Round Robin Performance", L.Teck Peow, T.Devadason, J.Siliquini, IEEE Region 10 Conference TENCON 2007, November 2007

[42] "On the Latency Bound of Deficit Round Robin", S.S.Kanhere, H.Sethu, Eleventh International Conference on Computer Communications and Networks, October 2002

[43] "Fair Scheduling Mechanisms with QoS Consideration for the IEEE 802.11e Wireless LAN", H-W.Ferng, H-Y.Liau, J-J.Huang, IEEE 65th Vehicular Technology Conference, April 2007

[44] "An Efficient Scheduling Algorithm for Packet Cellular Networks", H.Fattah, C.Leung, IEEE 56th Vehicular Technology Conference, Vol. 4, September 2002

[45] "A Dynamically Reconfigurable Queue Scheduler", C.Kachris, S.Vassiliadis, International Conference on Field Programmable Logic and Applications, August 2006

[46] "Differentiated MAI Control for Hybrid ARQ-II in CDMA Networks", W.Jiao, S.Li, IEEE Global Telecommunications Conference, December 2003

[47] "Extended DWRR Scheduling Algorithm in PFTS", D.Xu, X.Zhang, J.Zhao, IEEE Workshop on IP Operations and Management, 2004

[48] "Statistical Analysis of the Generalized Processor Sharing Scheduling Discipline", Z-L.Zhang, D.Towsley, J.Kurose, IEEE Journal on Selected Areas in Communications, Vol. 13, Issue 6, August 1995