Lab

3

Packet Scheduling

Teams: This lab may be completed in teams of 2 students (Teams of three or more are not permitted. All members receive the same grade).

Purpose of this lab:

Packet scheduling algorithms determine the order of packet transmission at the output link of a packet switch. The simplest and most widely used scheduling algorithm is FIFO (First-in-First-out), also called FCFS. Scheduling algorithms can become quite complex if the packet switch needs to differentiate traffic types.

Software Tools:

- The programming for this lab is done in Java and requires the use of *Java datagrams*.
- This labs uses traffic traces from Lab 1 and the leaky-bucket component from Lab 2.

What to turn in:

- Turn in a hard copy of all your answers to the questions in this lab, including the plots, hard copies of all your Java code, and the anonymous feedback form.

Version 2 (February 25, 2008)

Table of Content

Preparing for Lab 3

This lab requires the programs from Labs 1 and 2.

Comments

- **Quality of plots in lab report:** This lab asks you to produce plots for a lab report. It is important that the graphs are of high quality. All plots must be properly labeled. This includes that the units on the axes of all graphs are included, and that each plot has a header line that describes the content of the graph.

- **Extra credit:** Each part of this lab has an extra credit component (5% each).

- **Feedback:** To be able to improve the labs for future years, we collect data on the current lab experience. You must submit an anonymous feedback form for each lab. Please use the feedback form at the end of the lab, and return the form with your lab report.

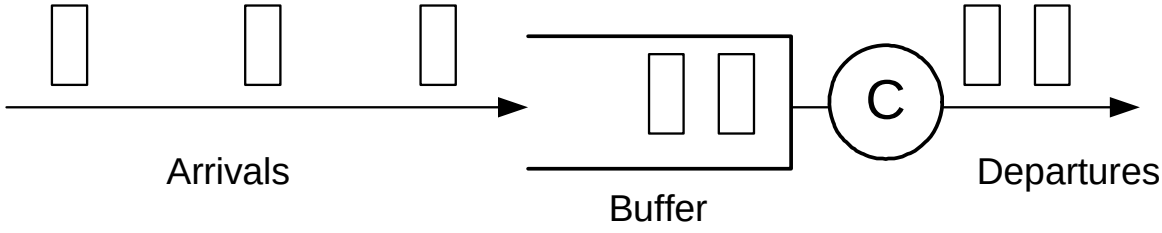
- **Java:** In the Unix lab, the default version of the Java installation is relatively old. To access a more recent version use the following commands:

    i. Compiling: /local/java/jdk1.5.0_09/bin/javac
    ii. Running: /local/java/jdk1.5.0_09/bin/java

Part 1. FIFO Scheduling

The first objective of this lab is to explore the backlog and delay at a link with a FIFO (First-In-First-Out) buffer, when the traffic load arriving to the link is increased.

The FIFO buffer operates at link capacity C=1 Mbps. The situation is illustrated in the figure, where packets are represented as rectangular boxes. With FIFO, also referred to as FCFS (First-Come-First-Served), packets are transmitted in the order of their arrival.



The goal of this exercise is to observe and measure the backlog and delay at a FIFO buffer when the load is varied. For traffic arrivals you will use the compound Poisson process.

Exercise 1.1 Traffic generator for compound Poisson traffic

Work with the traffic generator from Exercise 2.1 in Lab 2, which is based on the compound Poisson arrival process from Lab 1, where packet arrival events follow a Poisson process with rate λ = 1250 packets/sec, and the packet size has an exponential distribution with average size $1/\mu = 100$ Bytes. The average rate of this flow is 1 Mbps.

Download a trace with the Poisson data from:

http://www.comm.utoronto.ca/~jorg/teaching/ece466/labs/lab1/poisson3.data

- Consider the traffic generator that was built in Exercise 2.1 of Lab 2. Recall that the information in the trace file is re-scaled as follows:
    - The time values in the file are multiplied by a factor of 10;
    - The packet size values in the file are multiplied by a factor of 10.

  This results in a compound Poisson process with packet arrival rate λ = 125 packets/sec, and the packet size has an exponential distribution with average size $1/\mu$ =1000 Bytes. The average traffic rate is unchanged with this change, and remains 1 Mbps.

  At a link with 1 Mbps, the above Poisson source will generate an average load of 100% of the link capacity. The load of a link, also referred to as utilization and denoted by ρ, indicates the percentage of time that a work-conserving link will be busy (busy = transmitting a packet). The utilization is computed as follows:
  ρ = (Average packet arrival rate) x (average transmission time of packet)

  $= \lambda \times 1/(\mu C)$

- Add a feature to the code of your traffic generator that can re-scale the packet size to generate an average traffic rate of $N \cdot 0.1$ Mbps, where N = 1 (low load), N=5 (medium load), N=9 (high load).

- Test the correctness of the traffic generator, using the traffic sink from Part 2 in Lab 2.

Exercise 1.2 Implement a FIFO Scheduler

Build a FIFO scheduler that accepts arrivals from the traffic generator of the last exercise, and that transmits to a sink. The FIFO scheduler must satisfy the following requirements:

a. The FIFO scheduler must be able to receive a packet, while a packet is being transmitted. This can be done by using a separate thread for receiving packets.

b. The FIFO scheduler transmits an arriving packet immediately if no packet is in transmission. Otherwise, the packet is added to the buffer.

c. After completing the transmission of a packet, the transmitter selects the packet from the buffer with the earliest arrival time.

d. Set the maximum size of the buffer in the FIFO scheduler to 100 kB. If the available buffer size is too small for an arriving packet, the packet is discarded (and a message is displayed.)

- Determine the maximum rate at which your FIFO scheduler can transmit packets.

Exercise 1.3 Observing a FIFO Scheduler at different loads

Use the traffic generator to evaluate the FIFO scheduler with compound Poisson traffic at different loads.

- Use the added feature in the traffic generator from Exercise 1.1 and run the re-scaled Poisson trace file with an average rate of $N \cdot 0.1$ Mbps, where N = 1 (low load), N=5 (medium load), N=9 (high load).

- For each value of N, determine the following values:

    - Maximum backlog and waiting time in the buffer;

    - Average backlog and waiting time in the buffer;

    - Percentage of time that the FIFO scheduler is transmitting;

    - Percentage of time that a packet is waiting (i.e., in buffer and <u>not</u> in transmission).

    - Percentage of traffic that is discarded due to a buffer overflow.

- Present plots that show the above values as a function of N.

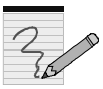
- Designate ranges of N, where the FIFO scheduler is in a regime of low load and high load. Justify your choice.

Exercise 1.4 (Optional, 5% extra credit) Unfairness in FIFO

A limitation of FIFO is that it cannot distinguish different traffic sources. Suppose that there are many low-bandwidth sources and a single traffic source that sends data at a very high rate. If the high-bandwidth source causes an overload at the link, then all traffic sources will experience packet losses due to buffer overflows. From the perspective of a low-bandwidth source, this seems unfair. (The low-bandwidth source would like to see that packet losses experienced at the buffer are proportional to the traffic rate of a source).

The following experiment tries to exhibit the unfairness issues of FIFO for two traffic sources.

- There are two traffic sources, which are each re-scaled compound Poisson sources as in Exercise 1.1:
    - Source 1 sends at an average rate of $N_1 \cdot 0.1$ Mbps.
    - Source 2 sends at an average rate of $N_2 \cdot 0.1$ Mbps.
- Both sources send traffic into a FIFO scheduler (with rate C=1 Mbps) and 100 kB of buffer.
    - Run a series of experiments where the load of the two sources is set to (N_1, N_2) with $N_1 = 5$ and $N_2 = 1, 5, 9$.
    - Record the average throughput (output rate) of each traffic source (Note: You must be able to keep track whether a packet transmission is due to Source 1 or Source 2).
    - Prepare a table that shows the average throughput values and interpret the result.
    - Is it possible to write a formula that predicts the throughput as a function of the arrival rate?



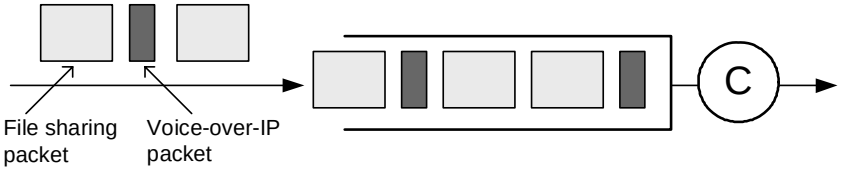
Lab Report:

Provide the plots and a discussion of the plots. Also include answers to the questions.

Part 2. Priority Scheduling

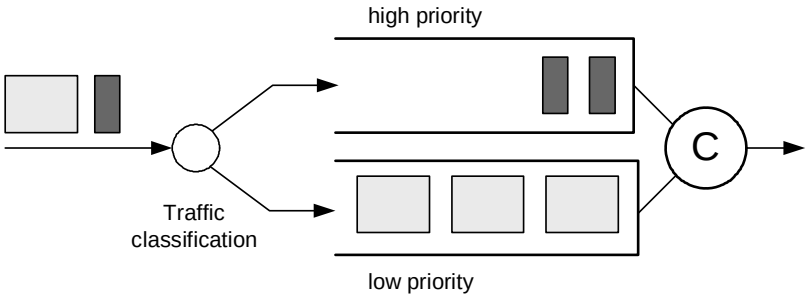
FIFO scheduling gives all traffic the same type of service. If the network carries traffic types with different characteristics and service requirements, a FIFO scheduling algorithm is not sufficient. To differentiate traffic types and give each type a different grade of service, more sophisticated scheduling algorithms are needed. One of these algorithms is the priority scheduling algorithm.

For example, consider a mix of file sharing traffic and voice (e.g., voice-over-IP) traffic: File sharing traffic is high-volume and transmitted in large packets, whereas voice-over-IP traffic has a relatively low data rate and is transmitted in short packets. If the traffic is handled by a FIFO scheduler, as shown below, then voice packets may experience a poor grade of service dependent on the on the arrivals of file sharing packets.



FIFO scheduler

A priority scheduler can improve the service given to voice packets by giving voice packets higher priority. A priority scheduler always selects the packet with the highest priority level for transmission.

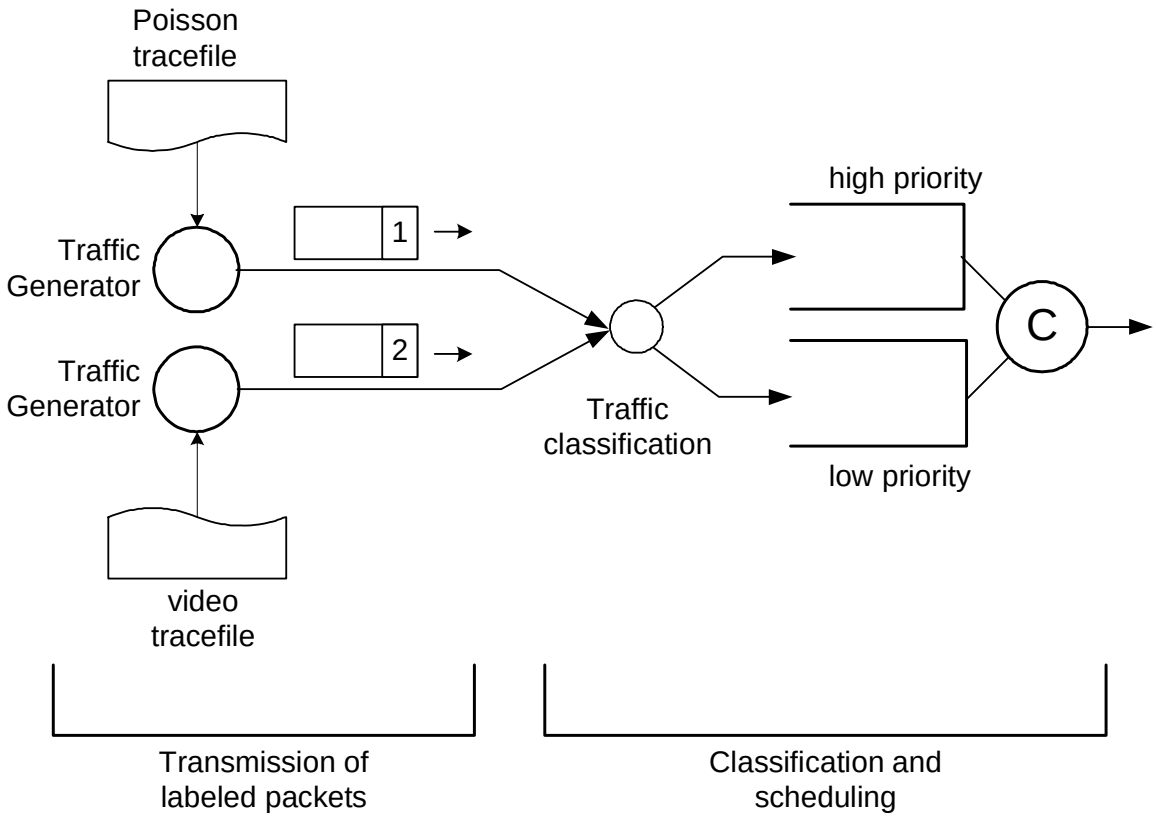


Priority scheduler

A priority scheduler assumes that incoming traffic can be mapped to a priority level. A traffic classification component of the scheduler uses an identifier in incoming packets to perform this mapping. The identifier can be based on source and destination addresses, application type application type (e.g., via the port number), or other information in packet headers.

Priority schedulers are also referred to as static priority (SP) or Head-of-Line (HOL) schedulers.

In this part of the lab, you will design and implement a priority scheduler with two priority levels, as shown in the Figure below. Traffic is transmitted to the scheduler from two sources: a compound Poisson source and a video source. The sources label packets with an identifier: "1" for packets from the Poisson source and "2" for packets from the video source. A traffic classifier at the priority scheduler reads the identifier. Packets with label "1" are handled as low priority, and packets with label "2" are handled as high priority packets.



Exercise 2.1 Transmission of labeled packets

Build a traffic generator as shown on the left hand side of the figure above. The requirements for the transmission of packets are as follows:

- Build a traffic generator for a video tracefile and for a Poisson tracefile. The transmissions of the video source and the Poisson source are performed by two distinct programs.

- The transmission of the video source that generates packets at a rate of 256 kbps. The tracefile for the video source can be downloaded from
  http://trace.eas.asu.edu/TRACE/pics/FrameTrace/h263/Verbose_Jurassic_256.dat

  The format of the file is shown below.

```
#Time [ms]   Frametype       Length [byte]
#_____________________________________________
      0              I         687
     40              P         345
    120             PB        7584
```

… … ….

- A traffic generator for the video source can be build by re-using the code from Exercise 1.2 and Exercise 2.1 from Lab 2. As in Lab 2, the maximum amount of data that can be put into a single packet is 1480 bytes. Frames exceeding this length are divided and transmitted in multiple packets.

- The transmission of the Poisson source is determined by the Poisson traffic generator built in Exercise 1.1 of this lab (Lab 3). The traffic generator must be able to run the re-scaled Poisson trace file with an average rate of $N \cdot 0.1$ Mbps, where N = 1 (low load), N=5 (medium load), N=9 (high load). .

- Before a packet is transmitted it must be labeled with an identifier, which is located in the first byte of the payload. The identifier is the number 0x01 for packets from the Poisson source and 0x02 for the video source.

- Packets are transmitted to a remote UDP port. Both sources transmit UDP datagrams to the same destination port on the same host (e.g., port 4444). You may use a traffic sink as build for Exercise 2.2 of Lab 2 for testing the implementation.

Exercise 2.2 Packet classification and priority scheduling

Build a traffic classification and scheduling component as shown on the right hand side of the above figure.

- The priority scheduler consists of two FIFO queues: one FIFO queue for high priority traffic and one FIFO queue for low priority traffic. Set the maximum buffer size of each FIFO queue to 100 kB.

- The priority scheduler always transmits a high-priority packet, if the high priority FIFO queue contains a packet. Low priority packets are selected for transmission only when there are no high priority packets.

- The transmission rate of the link is C=1 Mbps.

- The traffic classification component reads the first byte of the payload of an arriving packet and identifies the priority label. (The starting point for the traffic classification component can be the FIFO scheduler from Exercise 1.2.)

- Once classified, packets are assigned to the priority queues. Video traffic (with label “2”) is assigned to the high priority queue and Poisson traffic (with label “1”) is assigned to the low priority queue.

- If a new packet arrives when the link is idle (i.e., no packet is in transmission and no packet is waiting in the FIFO queues) the arriving packet is immediately transmitted. Otherwise, the packet is enqueued in the corresponding FIFO queue.

- The priority scheduler is work-conserving: As long as there is a packet waiting, the scheduler must transmit a packet.

- Packet transmissions is non-preemptive: Once the transmission of a packet has started, the transmission cannot be interrupted. In particular, when a low priority packet is in

transmission, an arriving high-priority packet must wait until the transmission is completed.

Test the program with the traffic generator from Exercise 2.1.

Exercise 2.3 Evaluation of the priority scheduler

Evaluate the priority scheduler with the traffic generator Exercise 2.1 using the following transmission scenarios:

- The video source transmits according to the data in the tracefile (see Exercise 2.1).
- As in Exercise 1.1, the Poisson source is re-scaled so that it transmits with an average rate of $N \cdot 0.1$ Mbps, where N = 1 (low load), N=5 (medium load), N=9 (high load).
- For each value of N, determine the following values for both high and low priority traffic:
    - Maximum backlog in the buffers;
    - Average backlog in the buffer;
    - Percentage of time that the FIFO scheduler is transmitting;
    - Percentage of time that a packet is waiting in the high (and low priority) queue.
- Present plots that show the above values as a function of N.
- Compare the outcome to Exercise 1.3.

Exercise 2.4 (Optional, 5% extra credit) Starvation in priority schedulers

A limitation of SP scheduling is that it always gives preference to high-priority scheduling. If the load from high-priority traffic is very high, it may completely pre-empt low-priority traffic from the link. This is referred to as *starvation*.

The following experiment tries to exhibit the starvation of low priority traffic. The experiment is similar to the last exercise of Part 1.

- Consider the previously build priority scheduler with two priority classes.
- There are two traffic sources, which are each re-scaled compound Poisson sources as in Exercise 1.1:
    - Source 1 transmits at an average rate of $N_1 \cdot 0.1$ Mbps.
    - Source 2 transmits at an average rate of $N_2 \cdot 0.1$ Mbps.

- Traffic from Source 1 is labelled with identifier “1” (low priority) and Source 2 is labelled with “2” (high priority).

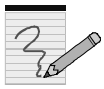
- Both sources send traffic to the priority scheduler (with rate C=1 Mbps) and 100 kB of buffer for each queue.

  - Run a series of experiments where the load of the two sources is set to (N_1, N_2) with $N_1 = 5$ and $N_2 = 1$, 5, and 9.

  - Record the average throughput (output rate) of each traffic source.

  - Prepare a table that shows the average throughput of high and low priority traffic and interpret the result.

  - Compare the outcome to Exercise 1.4.



Lab Report:

Provide the plots and a discussion of the plots. Also include answers to the questions.

Part 3. Weighted Round Robin (WRR) Scheduler

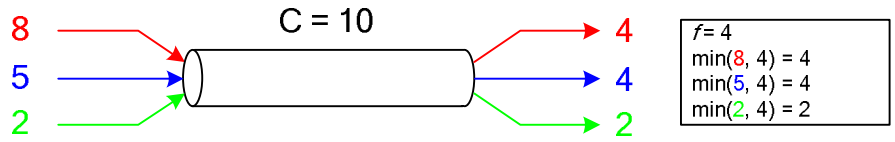
Many scheduling algorithms attempt to achieve a notion of **fairness** by regulating the fraction of link bandwidth allocated to each traffic source. The objectives of a "fair scheduler" are as follows: If the link is not overloaded, a traffic source should be able to transmit all of its traffic. If the link is overloaded, each traffic source obtains the same rate guarantee, called the *fair share*, with the following rules:

- If the traffic from a source is less than the fair share, it can transmit all its traffic;
- If the traffic from a source exceeds the fair share, it can transmit at a rate equal to the fair share.

The fair share depends on the number of active sources and their traffic rate. Suppose we have set of sources where the arrival rate of Source i is r_i, and a link with capacity C (bps). If the arrival rate exceeds the capacity, i.e., $\sum_i r_i \geq C$, then the fair share is the number f that satisfies the equation:

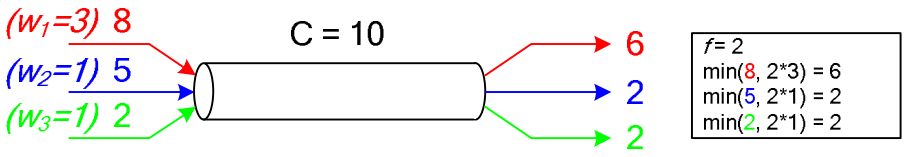
$$\sum_i \min(r_i, f) = C$$

As an example, suppose we have a link with capacity 10 Mbps, and the arrival rates of flows are r_1=8 Mbps, r_2= 6 Mbps, and r_3= 2 Mbps, then the fair share is f = 4 Mbps, resulting in an allocated rate is 4 Mbps for Source 1, 4 Mbps for Source 2, and 2 Mbps for Source 3.



Since different sources have different resource requirements, it is often desirable to associate a weight (w_i) with each source, and allocate bandwidth proportionally to the weights. In other words, a source that has a weight twice as large of second source should be able to obtain twice the bandwidth of the second source. With these weights, the fair share f at an overloaded link, i.e., $\sum_i r_i \geq C$ f $\times w_i) = C$ ined by solving .

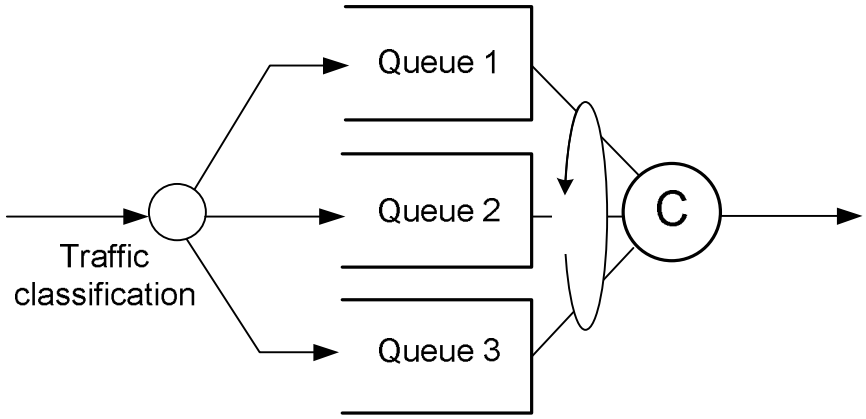
For example, using the previous example, and assigning weights w_1=3, w_2=w_3=1, the allocated rates are 6, 2, and 2 Mbps for the three sources.



A scheduling algorithm that realizes this scheme without weights is called Processor Sharing (PS) and with weights Generalized Processor Sharing (GPS). Both PS and GPS are idealized algorithms, since they treat traffic as a fluid. Realizing fairness in a packet network turns out to be quite hard, since packet sizes have different sizes (50 – 1500 bytes) and packet transmissions cannot be interrupted. Many commercial IP routers and Ethernet

switches (not the cheap ones!) implement scheduling algorithms that approximate the GPS scheduling algorithm.

A widely used (and easy to implement) scheduling algorithm that approximates GPA is the Weighted Round Robin (WRR) scheduler. The objective of this part of the lab is implementing and evaluating a WRR scheduling algorithm.

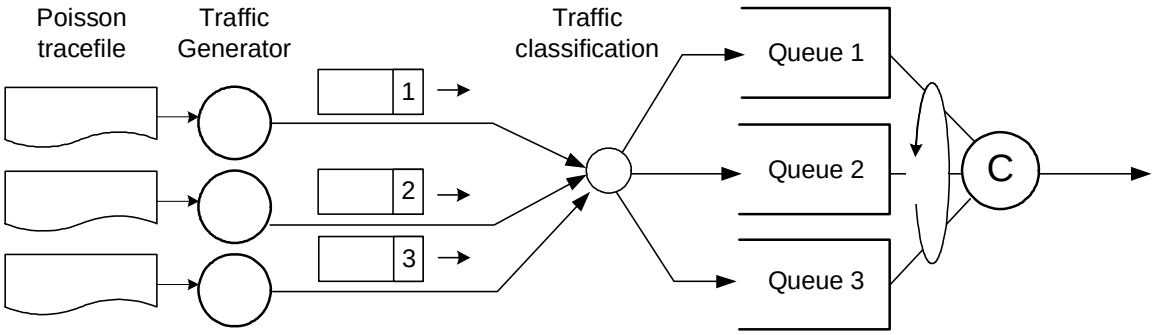


A **WRR scheduler**, illustrated in the figure above, operates as follows:

- *The scheduler has multiple FIFO queues. A traffic classification unit assigns an incoming packet to one of the FIFO queues.*
- *The WRR scheduler operates in "rounds". In each round the scheduler visits each queue in a round-robin (sic!) fashion, starting with Queue 1. During each visit of a queue one or more packets may be serviced.*
- *The WRR assumes that one can estimate (or know) the average packet size of the arrivals to Queue i, denoted by L_i.*
- *The WRR calculates the number of packets to be served in each round:*
    - *For each Queue i: $x_i = w_i / L_i$*
    - *$x = min_i \{ x_i \}$*
    - *For each Queue i: $packets\_per\_round_i = x_i / x$*
- *Once all packets of a round are transmitted or if no more packets are left, the scheduler visits the next queue.*

Exercise 3.1 Build a WRR scheduler

Build a WRR scheduler as described above, that supports at least three queues. The WRR will serve Poisson traffic sources as used in Parts 1 and 2 of this lab.



- There are three traffic generators that each transmit re-scaled Poisson traffic as done in Exercise 1.1. Each Poisson source is re-scaled so that it transmits with an average rate of $N \cdot 0.1$ Mbps, where N = 1 (low load), N=5 (medium load), N=9 (high load).

- Each transmitted packet is labelled with "1", "2" or "3" as done in Part 2 of this lab. The label is one byte long. The traffic classification component reads the first byte of the payload of an arriving packet, and adds the packet to the queue (Packets with label "1" are associated with Queue 1, etc.). If no packet is in transmission or in the queue, and arriving packet is transmitted immediately.

- The transmission rate of the link is C=1 Mbps. The maximum buffer size of each queue is 100 kB. An arrival that cannot be stored in the queue is discarded.

Once the implementation is completed and tested, move on to the evaluation.

Exercise 3.2 Evaluation of a WRR scheduler: Equal weights

Evaluate the WRR scheduler with three Poisson sources.

- The weight N of the traffic generators is set to N=8 in for the first source, N=6 for the second source and N=2 for the third source.

- Set the weights of the queues to $w_1=w_2=w_3=1$.

- The average packet size of a source is set to $N \cdot 100$ bytes, where N is the weight.

Note: Compare this scenario to the first figure of Part 3. The average load on the link is 1.5 Mbps, i.e., the link is overloaded. We expect that the bandwidth at the link is shared among the sources at a ratio of 4:4:2.

- Prepare plots that show the number of packet transmissions and the number of transmitted bytes from a particular source (y-axis) as a function of time (x-axis). Provide one plot for each source.
  Select a reasonable time scale for the x-axis, e.g., a time scale of 10 ms per data point.

- Compare the plots with the theoretically expected values of a PS scheduler.

Exercise 3.3 Evaluation of a WRR scheduler: Different weights

This exercise re-creates a transmission scenario as in the second figure of Part 3.

- Evaluate the WRR scheduler with three Poisson sources. As before, the weight N of the traffic generators is set to N=8 in for the first source, N=6 for the second source and N=2 for the third source.

- Set the weights of the queues to w_1=3 and w_2=w_3=1.

Note: Compare this scenario to the second figure of Part 3. We expect that the bandwidth at the link is shared among the sources at a ratio of 6:2:2.

- Prepare plots that show the number of packet transmissions and the number of transmitted bytes from a particular source (y-axis) as a function of time (x-axis). Provide one plot for each source.

- Compare the plots with the theoretically expected values of a GPS scheduler.

Exercise 3.4 (Optional, 5% extra credit) No Unfairness and no Starvation in WRR

A limitation of SP scheduling is that it always gives preference to high-priority scheduling. If the load from high-priority traffic is very high, it may completely pre-empt low-priority traffic from the link. This is referred to as *starvation*.

The following experiment tries to show that WRR does not suffer from the problems of FIFO (unfair) and SP (starvation). The experiment retraces the steps of Exercises 1.4 and 2.4.

- Consider the WRR scheduler from above with rate C=1 Mbps and 100 kB of buffer for each queue.

- There are two traffic sources, which are each re-scaled compound Poisson sources as in Exercise 1.1:

    - Source 1 transmits at an average rate of $N_1 \cdot 0.1$ Mbps.

    - Source 2 transmits at an average rate of $N_2 \cdot 0.1$ Mbps.

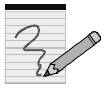
    - Traffic from Source 1 is labelled with identifier “1” (low priority) and Source 2 is labelled with “2” (high priority).

    - Both sources are assigned the same weight at the WRR scheduler ($w_1=w_2=1$).

- Run a series of experiments where the load of the two sources is set to (N_1, N_2) with N_1 = 5 and N_2 = 5, 9, 15.

    - Record the average throughput (output rate) of each traffic source.

- Prepare a table that shows the average throughput of the two sources and interpret the result.
- Compare the outcome to Exercises 1.4 and 2.4.



Lab Report:

Provide the plots and a discussion of the plots. Also include answers to the questions.

Feedback Form for Lab 3

- Complete this feedback form at the completion of the lab exercises and submit the form when submitting your lab report.
- The feedback is anonymous. **Do not put your name on this form** and keep it separate from your lab report.
- For each exercise, please record the following:

	Difficulty (-2,-1,0,1,2) -2 = too easy 0 = just fine 2 = too hard	**Interest Level (-2,-1,0,1,2)** -2 = low interest 0 = just fine 2 = high interest	**Time to complete (minutes)**
Part 1. FIFO Scheduling			
Part 2. Priority Scheduling			
Part 3. Weighted Round Robin (WRR) Scheduler			

Please answer the following questions:

- What did you like about this lab?
- What did you dislike about this lab?
- Make a suggestion to improve the lab.