

Question 2:

- a- In this case, the algorithm EDF is better than BD.

The reason is that when EDF has to process a packet, the one with the lowest slack will be transmitted. When we reduce the slack by one, the chances are higher to get slack 0.

So EDF will retrieve the lowest slacks packets, which means that there are less chances for EDF to drop packets than BD.

For example, let's represent a queue with slacks value : 1,2,3,4

After reducing by one we get : 0,1,2,3, and then EDF will process the packet with slack 1.

On the next round, EDF we get : 0,0,1,2, EDF will process the packet with slack 1, same for the last round. Overall 1 packet was dropped.

But BD on the other hand will drop two packets :

Queue at the beginning : 1,2,3,4 and after reducing slack by one:

First round : 0,1,2,3 : 3 will be transmitted and 0 is dropped.

Next round : reducing slack by one we get : 0,0,1,2 : another packet was dropped, because BD retrieved 3 instead of 1.

So BD dropped two packets, and EDF dropped one.

In general, EDF will "save" more packets than BD, that's why EDF is better than BD when values are the same but slacks are heterogenic.

- b- In this case, BD will be better than EDF.

The reason is that BD will always try to transmit the packets with the highest value.

Which means that BD tends to accumulate higher values than EDF.

EDF will look at the lowest slack, and in the case where the slack is low, then it doesn't matter what is the value, the lowest slack packet will be processed.

On the other hand, BD will always accumulate the packets with the highest values, so BD will give better results and won't drop high values packets.

- c- BD should be performed when we want to transmit large packets, in other words, the packets with the highest values.

On the contrary, EDF should be performed when we want to send packets and get a minimum packet loss, the value doesn't matter. EDF sends data based on minimum slack, therefore, no need to worry about how much is sent but rather avoid packet dropping.

- d- Other policies for this exercise would be :

-Drop detection algorithm :

This method is used to detect unsent packets by sending acknowledgment when the packets were received, if the packet is not acknowledged fast enough, we assume that the packet was dropped.

The advantage is that we can control dropped packets and resend them if they were not sent.

The disadvantage is that the detection of dropped packets in a certain frame of time makes the algorithm more complex and the right manipulations have to be done on dropped packets in order to retransmit them shortly and properly while not getting confused.

-Least slack time scheduling :

Assign priority to jobs based on slack time, The smaller the slack time, the higher the priority. (Continue to the next page please).

Disadvantage : More complex, requires knowledge of execution times and deadlines.

Advantages : Always produce a feasible schedule if one exists – Constraints: on a single processor, as long as pre-emption is allowed and jobs do not contend for resources