CS 161

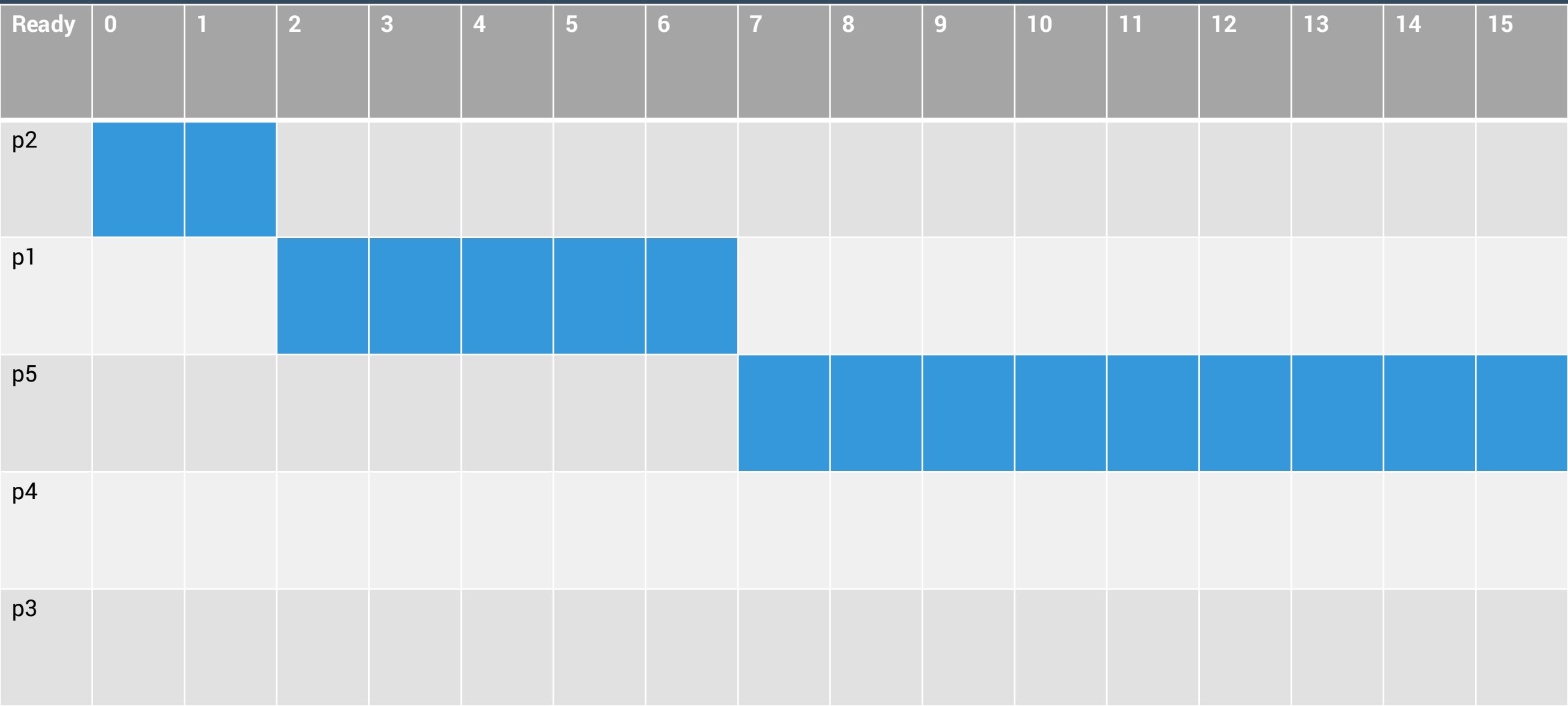
An Analysis of the Dynamic Average Burst CPU Scheduling Algorithm

Constantino, Cu, Yumang

How does it work?

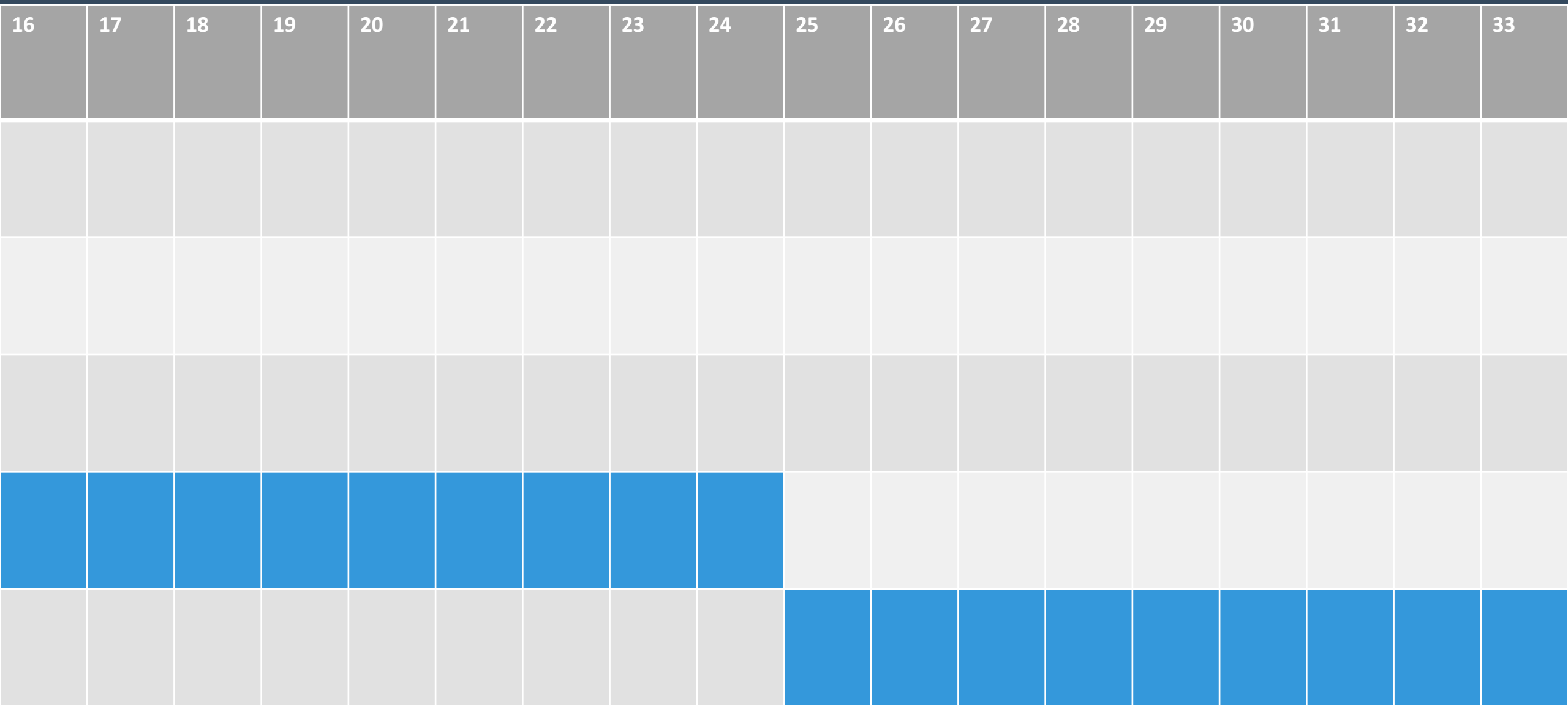
Round 1

Average : 9



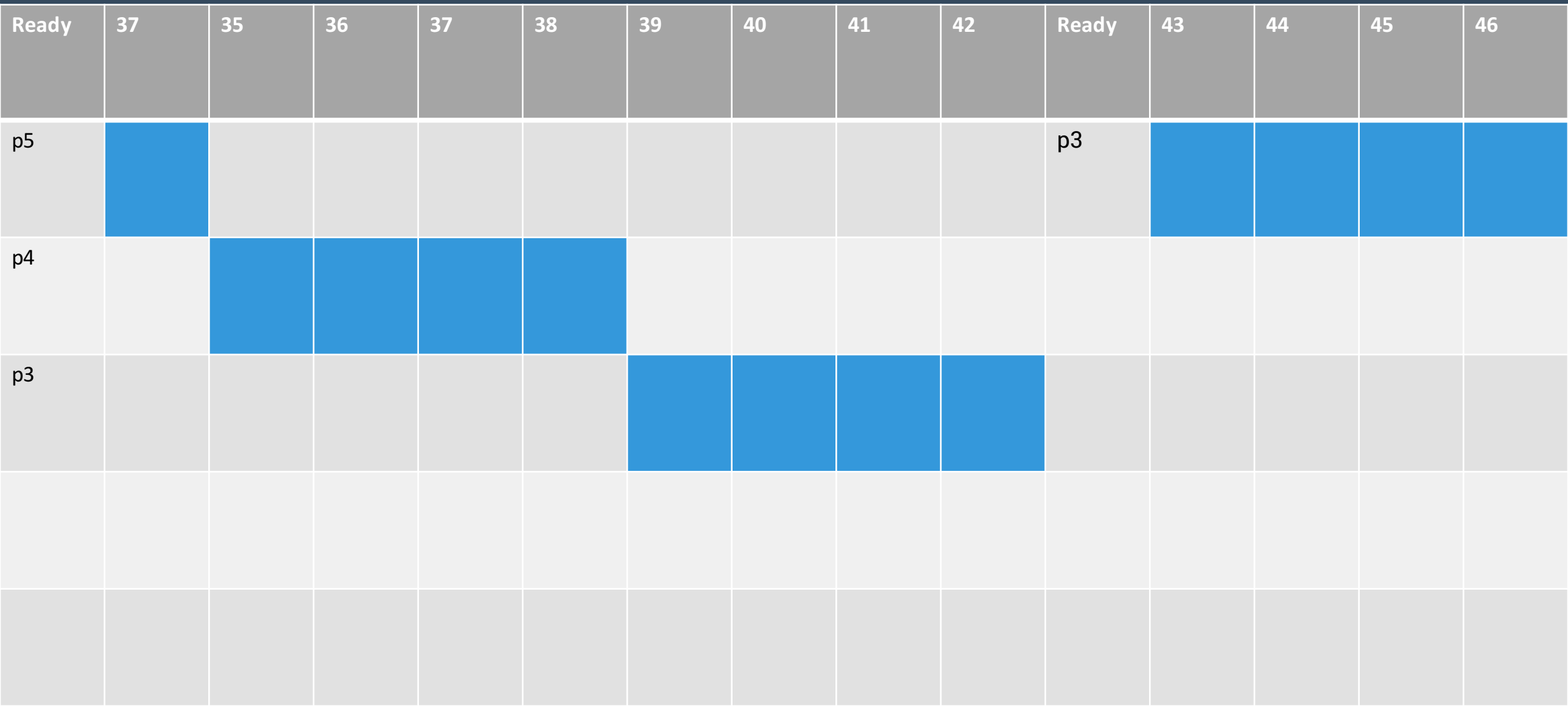
Round 1

Average : 9



Round 2 & 3

Average : 4



Difficulties and Issues

Difficulties and issues

- x No Priority
- x No Response Time
- = Same CPU bursts

Approaches used to test the algorithm

Deterministic model

- Number of context switch
- Average waiting time
- Average turnaround time

Case 1. Zero arrival time with ascending order of processes

Case 2. Zero arrival time with descending order of processes

Case 3. Zero arrival time with Random order of processes

Case 4. Without zero arrival time with ascending order of processes.

Case 5. Without zero arrival time with descending order of processes.

Case 6. Without zero arrival time with random order of processes

Performance vs Classic algorithms

Results

	With Arrival Time 0			
	Context Switch			
	Ascending burst time	Descending burst time	Random burst time	Total
RR	16	15	17	48
DQRR	7	7	7	21
IRRVQ	14	14	14	42
SARR	7	7	4	18
RP-5	11	11	11	33
MRR	8	7	8	23
DBRR	7	7	7	21

Case 1

	With Arrival Time 0			
	Turnaroud Time			
	Ascending burst time	Descending burst time	Random burst time	Total
RR	261.40	274.00	327.00	862.40
DQRR	231.60	209.40	274.40	715.40
IRRVQ	234.40	206.60	274.80	715.80
SARR	188.40	250.40	259.20	698.00
RP-5	236.40	289.40	319.40	845.20
MRR	193.80	171.40	250.20	615.40
DBRR	190.20	170.20	223.20	583.60

Case 2

	With Arrival Time 0			
	Waiting Time			
	Ascending burst time	Descending burst time	Random burst time	Total
RR	192.00	209.40	245.40	646.80
DQRR	162.20	144.80	192.80	499.80
IRRVQ	165.00	142.00	193.20	500.20
SARR	119.00	185.80	177.60	482.40
RP-5	167.00	224.80	237.80	629.60
MRR	124.40	106.80	168.60	399.80
DBRR	120.80	105.60	141.60	368.00

Case 3

Results

	Without 0 Arrival Time			
	Context Switch			
	Ascending burst time	Descending burst time	Random burst time	Total
RR	15	13	13	41
DQRR	7	7	7	21
IRRVQ	10	10	10	30
SARR	7	7	8	22
RP-5	8	8	8	24
MRR	7	7	8	22
DBRR	7	7	7	21

Case 4

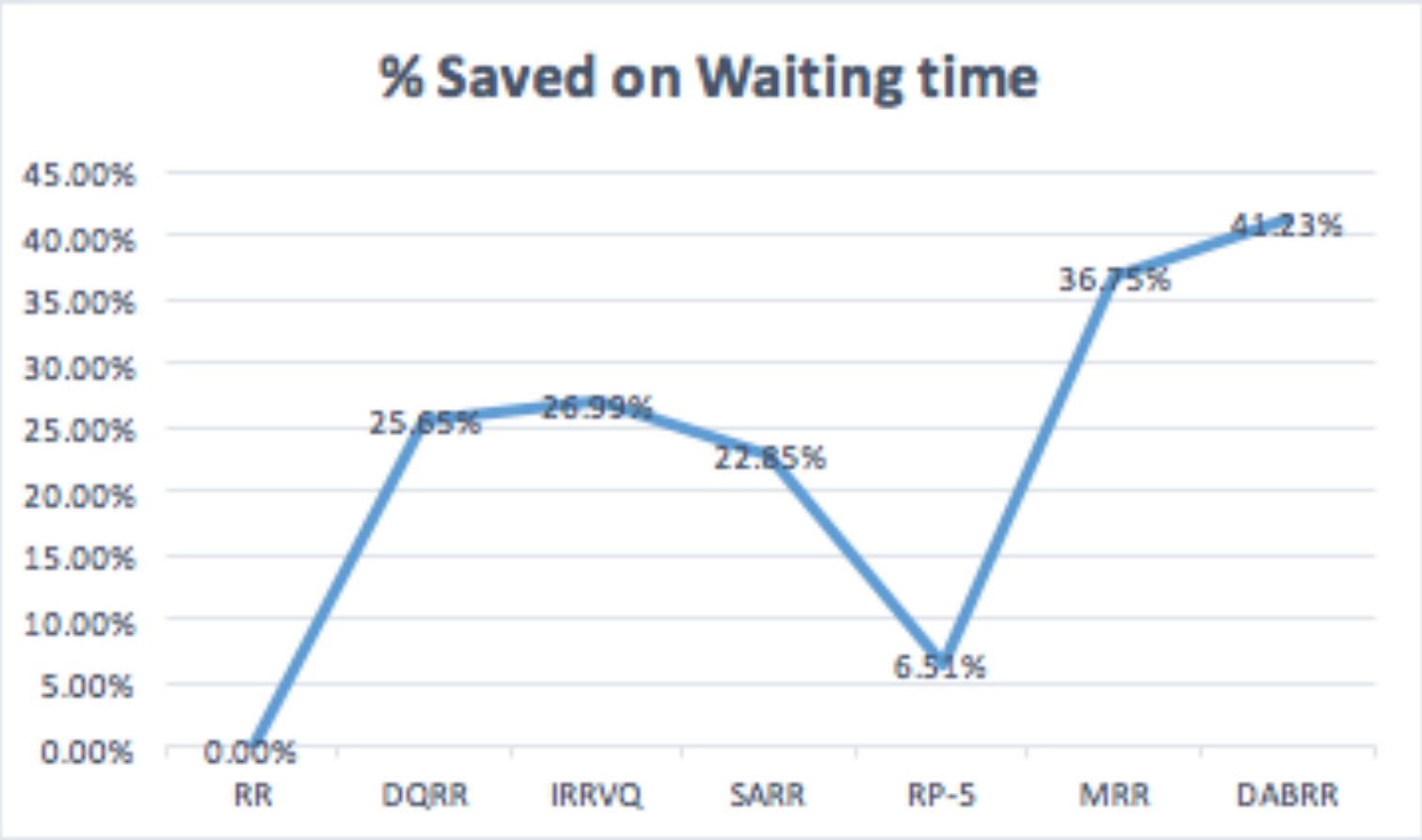
	Without 0 Arrival Time			
	Waiting Time			
	Ascending burst time	Descending burst time	Random burst time	Total
RR	144.40	191.00	173.20	508.60
DQRR	107.20	138.40	113.60	359.20
IRRVQ	98.20	133.80	111.40	343.40
SARR	88.00	172.40	148.60	409.00
RP-5	104.40	197.00	149.20	450.60
MRR	90.00	124.60	116.40	331.00
DBRR	88.20	125.00	97.80	311.00

Case 5

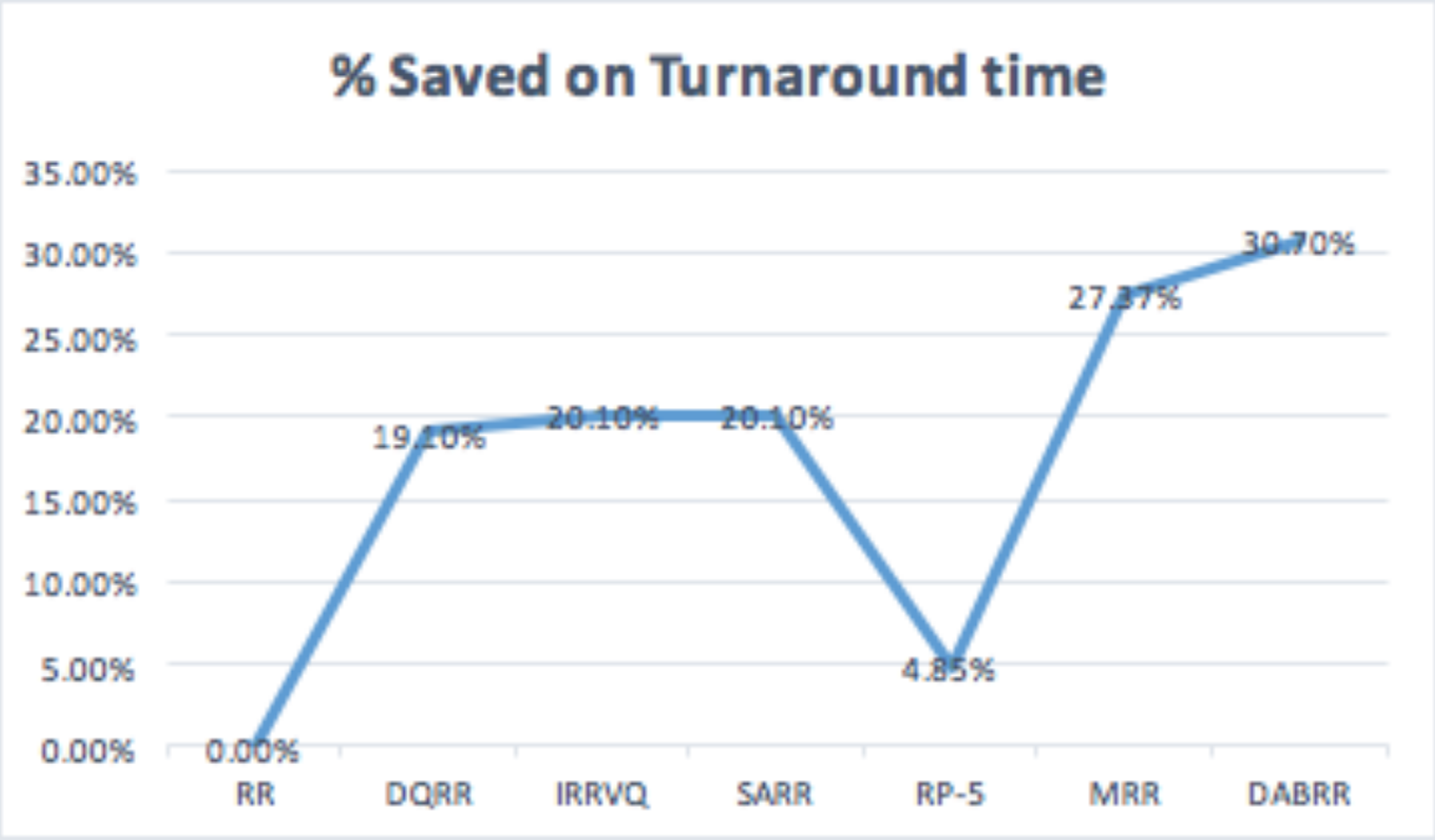
	Without 0 Arrival Time			
	Turnaround Time			
	Ascending burst time	Descending burst time	Random burst time	Total
RR	205.60	250.80	232.80	689.20
DQRR	168.40	198.20	173.20	539.80
IRRVQ	159.40	193.60	171.00	524.00
SARR	149.20	232.20	208.20	589.60
RP-5	165.60	256.80	208.80	631.20
MRR	151.20	184.40	176.00	511.60
DBRR	149.40	184.80	157.40	491.60

Case 6

Results


% Saved on Waiting time
45.00%
40.00%
35.00%
30.00%
25.00%
20.00%
15.00%
10.00%
5.00%
0.00%
0.00%
25.65%
26.99%
22.85%
6.51%
36.75%
41.23%
RR
DQRR
IRRVQ
SARR
RP-5
MRR
DABRR

Results


% Saved on Turnaround time
35.00%
30.00%
25.00%
20.00%
15.00%
10.00%
5.00%
0.00%
0.00%
19.10%
20.10%
20.10%
4.85%
27.37%
30.70%
RR
DQRR
IRRVQ
SARR
RP-5
MRR
DABRR

Advantages and Disadvantages

Results

Advantages

- Context switches ⬇
- Average Waiting Time ⬇
- Average Turnaround Time ⬇

Disadvantages

- Priority not addressed ✗
- Possibility of starvation !

Application in Uni- and Multiprocessor threads

Application in Uni- and Multiprocessor Threads

Uniprocessor

The application is round robin and if a process is still not done, it goes to the next round

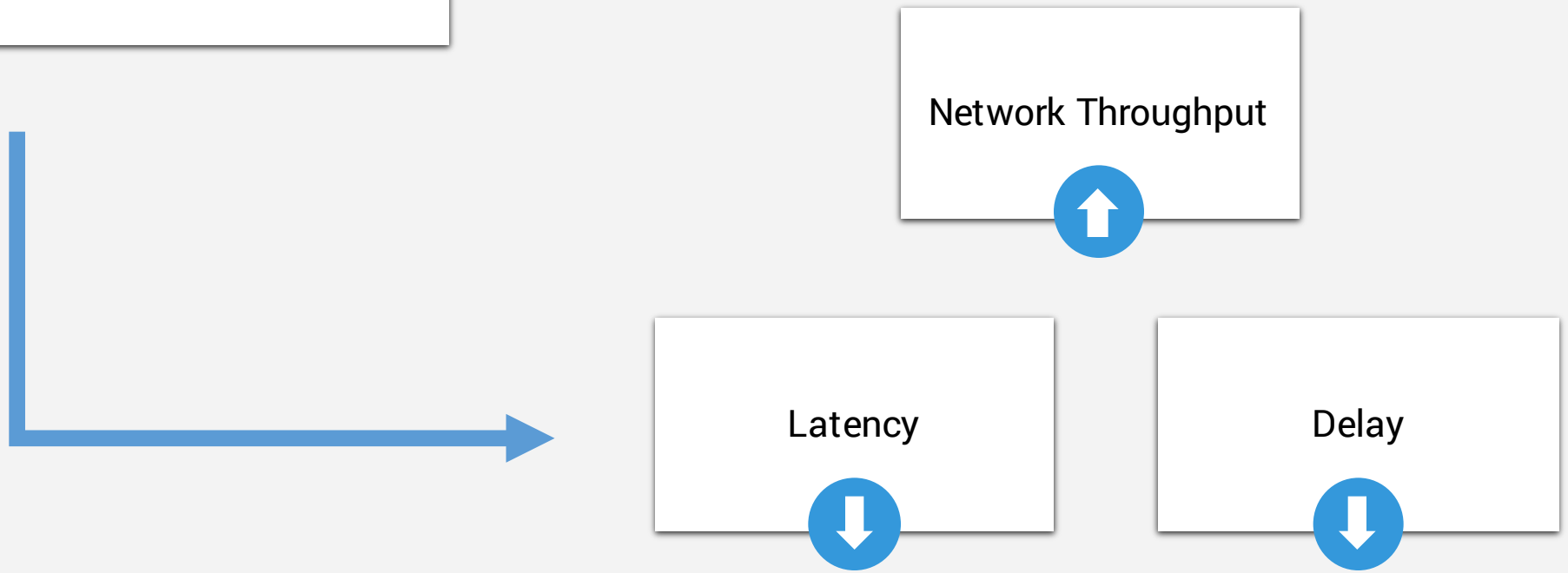
Multiprocessor

- Complicated to apply because of multiple job schedulers
- Different quantum times are to be calculated for each CPU
- In heterogeneous system, the CPU's have communicate to each other
- In homogenous system, the CPU's can perform load sharing

MIS application

Network Packet Scheduling

The network scheduler is an arbiter program in the packet switching communication network which manages the sequence of network packets both in receive and transmit queues. It is the network scheduler who decides which network packet to forward to, next, from the buffer.



End