**Probability of a Synchronous Control System**

Khaled Mohammed Saifuddin

Muhammad Ifte Khairul Islam

Farhan Tanvir

There are n slots in the block. We assume that the arrival rate is λ and service rate is μ. We are proposing a system where there will be no system loss. In our proposed system, no slots will be purged; all slots will be utilized. For this purpose, we are planning to use a **secondary block**. **Secondary block** will store slots whenever main block is full and provides service to its slots. Secondary block will have same maximum capacity as main block.

**Description of events:**

Considering the workflow of the whole system, we can say that there are 4 types of events.

* **Arrival of slots**: At the beginning or when time=0, main block will have no slots. Slots will arrive to the main block at arrival rate λ.

Slots arriving at arrival

rate λ

Main block Secondary block (empty)

* **Service:** When block will be filled up with n slots, it will start servicing the slots at service rate μ.

providing service

at service rate μ

Main block (filled up with n slots) Secondary block (empty)

* **Arrival of slots to main block from secondary block:** When main block will process slots, no arrived slots will be queued into the main block. Instead, arrived slots will enter into secondary block at arrival rate λ. After main block has processed its blocks, slots from secondary block will be queued into main block. We assume that the process of slots being queued into main block from secondary block takes 1 second each time.

Processing Slots arriving at λ

Slots at μ

Main block (processing slots) Secondary slot (slots arriving)

Main block (after processing) Secondary block

Main block (Slots have arrived Secondary block

from secondary block to main block)

* **Arrival of slots to main block (at arrival rate λ and takes place** **after slots are queued from secondary block):** Slots from secondary block have been queued into main block. Afterwards, remaining slots of the block will be filled at arrival rate λ.

First arrival of slots, the first mentioned event, will take place once and only at the beginning. Events mentioned after first arrival of slot, which are Service, Arrival of slots to main block(from secondary block) and arrival of slots to main block(at arrival rate λ), will take place in cycle.

**Calculation of each event’s operational time and probability:**

We will calculate necessary terms for each event. These terms will lead us to determining probability of I slots in the blocks. For most of the events, the necessary terms are operational time and probability.

* **Terms of Event- arrival of slots:**

Slots will be queued into main block. Slots can appear naturally into the system. Slots from secondary block have been queued into main block. Arrival event will be applicable for remaining slots in the main block. remaining slots of the block will be filled at arrival rate λ.

We have calculated total arrival time (total time required for the main block to be filled up with n slots).

TimeArrival =

where m = number of times slots have been queued into main block

and λ = arrival rate

TimeArrival(i) =

where i = 0…first arrival event’s n.

= number of times i-th slot has been queued into main block

* **Terms of Event- Service:**

When block will be filled up with n slots, it will start servicing the slots at service rate μ. Since all slots are processed at equal service rate and all service events are applicable for all slots, i slots are equiprobable in case of service events. We have taken into account number of service event here. All service events take the same time to complete. We have calculated total service time(total time required for the main block to process all slots) and probability of main block to have i slots during this event(Service).

TimeService(i) =

Where,

=

μ = service rate

n = number of slots

* **Terms of Events- Arrival of slots to main block from secondary block:**

When main block will process slots, no arrived slots will be queued into the block. Instead, arrived slots will enter into secondary block λ. After main block has processed its blocks, slots from secondary block will be queued into main block. We assume that the process of slots queued into main block from secondary block takes 1 second each time. We calculate the number of slots queued into secondary block during this time.

TimeQueuing(i) =

Where,

ℼ= Count (total i slots have queued into main block)

ℽ= Count (Events when slots queued into main block)

**Probability calculation:**

**Main block having no slots:** dependent on product of arrival rate and service time

Applying all these, we can write the probability equation as –

p[i]= \* +

Where,

Arrival\_Time ℼ =

Service\_Time T𝝁 =

Queuing time γ= Number of times slots queued from secondary to main block

System\_Time Tϭ = Arrival\_Time + Service\_Time + Queuing Time

𝜶i=

βi= Number of times i number of slots queued from secondary block

N=

Here, we assume that each slot takes 1s to move from secondary block to primary block

Here, Σp[i] = 1

A figure of our system is attached below

A close up of text on a white background

Description automatically generated

**Our proposed solution to problem-2**

In this problem, n will be adaptive. We assume that the arrival rate is λ and service rate is μ. We are proposing a system where there will be no system loss. In our proposed system, no slots will be purged; all slots will be utilized. For this purpose, we are planning to use a **secondary block**. **Secondary block** will store slots whenever main block is full and provides service to its slots. Secondary block has the storage capacity to store more blocks than main block.

**Description of events:**

Considering the workflow of the whole system, we can say that there are 4 types of events.

* **Arrival of slots**: For the first arrival, we will assume a value of n. At the beginning or when time=0, main block will have no slots. Slots will arrive to the main block at arrival rate λ.

Slots arriving at arrival

rate λ

Main block Secondary block (empty)

* **Service:** When block will be filled up with n slots, it will start servicing the slots at service rate μ

providing service

at service rate μ

Main block (filled up with n slots) Secondary block (empty)

* **Arrival of slots to main block from secondary block:** When main block will process slots, no arrived slots will be queued into the main block. Instead, arrived slots will enter into secondary block at arrival rate λ. After main block has processed its blocks, all slots from secondary block will be queued into main block. We assume that the process of slots being queued into main block from secondary block takes 1 second each time.

Processing slots arriving at λ

Slots at μ

Main block (processing slots) Secondary slot (slots arriving)

Main block (after processing) Secondary block

Main block (Slots have arrived Secondary block

from secondary block to main block)

* **Arrival of slots to main block(at arrival rate λ and takes place** **after slots are queued from secondary block):** Slots from secondary block have been queued into main block.

This is a critical part of our solution. Now, the number of slots, that had been in secondary block, will be the new value of n. After all slots are being queued into main block, main block will start processing slots. This way, the value of n will keep changing in each cycle according to number of slots in secondary block. Changing values of n can be more or less than the first assumed value of n.

First arrival of slots, the first mentioned event, will take place once and only at the beginning. Events mentioned after first arrival of slot, which are Service, Arrival of slots to main block(from secondary block) and arrival of slots to main block(at arrival rate λ), will take place in cycle.

**Calculation of each event’s operational time and probability:**

We will calculate necessary terms for each events. This terms will lead us to determining probability of I slots in the blocks. For most of the events, the necessary terms are operational time.

* **Terms of Event- arrival of slots:**

Slots will be queued into main block. Slots can appear naturally into the system. Slots from secondary block have been queued into main block.

We have calculated first arrival time(total time required for the main block to be filled up with n slots) and time when main block has I slots during first arrival event. This variables will only be applicable to 0…first arrival event’s n.

TimeArrival =

where n = number of slots

and λ = arrival rate

TimeArrival(i) =

where i = 0…first arrival event’s n

= number of times i-th slot has been queued into main block

* **Terms of Event- Service:**

Main block will start processing when main block is filled up with n slots. In other cases, main slot will start processing slots when slots are being queued from secondary block to main block. The value of n is adaptive. N changes from time to time. However, all slots are processed at equal service rate. So, we have to take to into account the total number of arrived slots to obtain accurate service time. All service events may not same time to complete since the value of n varies. So, the time a slot may spend in main block may vary from one slot to another. We have calculated total service time (total time required for the main block to process all slots) and time when main block has i slots during service event.

TimeService =

Timei\_Service(i) =

where μ = service rate

ℯ= Total number of serviced slots

ṫi= Number of time i-th slot has received service

* **Terms of Events- Arrival of slots to main block from secondary block:**

In our system, most arrival events will be slots being queued from secondary block to main block. To keep n adaptive, we are queuing slots from secondary block to main block and number of slots being queued from secondary block to main block will be our new n. Then, Main block will start processing the slots. In these events, no naturally arrived slots will be queued into main block. Naturally arrived slots will be queued into secondary block.We assume that the process of slots queued into main block from secondary block takes 1 second each time. We calculate the number of slots queued into secondary block during this time.

TimeQueuing(i) =

ℼ= Count (total i slots have queued into main block)

ℽ= Count (Events when slots queued into main block)

**Probability calculation:**

**Main block having no slots:** dependent on product of arrival rate and service time

Applying all these, we can write the probability equation as –

p[i]= \* +

Where,

Arrival time, ℼ =

Service time, T𝝁 =

System time, Tϭ= Arrival\_Time + Service\_Time + Queuing Time

𝜶i= No. of times i-th slot has been queued into main block

= No. of times i-th slot has been serviced

βi= No. of times i number of slots queued from secondary block

γ= No. of times slots queued from secondary to main block

Here, we assume that each slot takes 1s to move from secondary block to primary block

Here, Σp[i] = 1

A figure of our system is attached below

A screenshot of a cell phone

Description automatically generated

**Our proposed solution to problem-3**

In this problem, total number of slots in main block, n will be static as well as adaptive. The value of n has to be selected by the program. We are trying to implement a mechanism so that our program will choose n based on arrival rate, service rate and maximum allowed slots in a second. During service of events, arrived slots will be queued into secondary block. If number of slots in secondary block is greater than total number of slots in main block n, then n will be changed to number of slots in secondary blocks. However, after processing these slots, n will be reverted back to its old value. This way, n will be static as well as adaptive.

We assume that the arrival rate is λ and service rate is μ. We are proposing a system where there will be no system loss. In our proposed system, no slots will be purged; all slots will be utilized. For this purpose, we are planning to use a **secondary block**. **Secondary block** will store slots whenever main block is full and provides service to its slots. Secondary block has the storage capacity to store more blocks than main block.

Our goal is to provide instant service to slots. If instant service can’t be provided, we have to ensure instant service to most possible slots. And if any slot is in secondary block, waiting time for these slots should be reduced. However, in no case, any slots will be purged/wasted. We will utilize and provide service to each and every arrived slot. For this reason, service rate will be significantly higher than arrival rate.

In our system, there will be a limit to maximum arrived slots at a timeframe(second/minute) in a second.

We will consider few options for choosing n. The options are

* arrival rate
* maximum limit of arrived slots per second
* average of arrival rate and maximum limit of arrived slots
* service rate
* average of service rate and maximum limit of arrived slots

We have calculated approximate system time and approximate slots in secondary block for each of these options. Afterwards, we compare approximate system time and approximate secondary block size for each of these options. Whichever option has approximated lowest system time and secondary block size, block size is set to the option value. we give priority to lowest secondary block size from system time and secondary block size. if we can't find a clear winner considering both system time and secondary block size, we will choose whichever option has lowest secondary block size. The objective of the problem is slots should be processed as soon as they arrive. In order to do that, secondary slots should have fewer slots.

To sum up, the advantages of using this value of n is we can utilize space, provide instant service to slots, fewer slots have to go through waiting time, reduce waiting time for slots and most importantly, no slots will be wasted.

**Description of events:**

Considering the workflow of the whole system, we can say that there are 4 types of events.

* **Arrival of slots**: For the first arrival, we will assume a value of n. At the beginning or when time=0, main block will have no slots. Slots will arrive to the main block at arrival rate λ.

Slots arriving at arrival

rate λ

Main

Main block Secondary block

* **Service:** When slots will arrive at main block, it will start servicing the slots at service rate μ.

providing service

at service rate μ

Main block (filled up with n slots) Secondary block (empty)

* **Arrival of slots to main block from secondary block:** When main block will process slots, no arrived slots will be queued into the main block. Instead, arrived slots will enter into secondary block at arrival rate λ. After main block has processed its blocks, all slots from secondary block will be queued into main block. We assume that the process of slots being queued into main block from secondary block takes 1 second each time.

Processing slots arriving at λ

Slots at μ

Main block (processing slots) Secondary slot (slots arriving)

Main block (after processing) Secondary block

Main block (Slots have arrived Secondary block

from secondary block to main block)

* **Arrival of slots to main block (at arrival rate λ and takes place** **after slots are queued from secondary block):** Slots from secondary block have been queued into main block.

**During service of events, arrived slots will be queued into secondary block. If number of slots in secondary block is greater than n, then n will be changed to number of slots in secondary blocks. However, after processing these slots, n will be reverted back to its old value. This way, n will be static as well as adaptive.**

**Calculation of each event’s operational time and probability:**

We will calculate necessary terms for each event. These terms will lead us to determining probability of I slots in the blocks. For most of the events, the necessary terms are operational time and probability.

* **Terms of Event- arrival of slots:**

Slots will be queued into main block. Slots can appear naturally into the system. Slots from secondary block have been queued into main block.

We have calculated total arrival time(total time required for the main block to be filled up with n slots) and probability of main block to have i slots during this event.

TimeArrival =

where m = number of times slots have been queued into main block

and λ = arrival rate

TimeArrival(i) =

where i = 0…first arrival event’s n.

= number of times i-th slot has been queued into main block

* **Terms of Event- Service:**

Main block will start processing when slots arrive. In other cases, main slot will start processing slots when slots are being queued from secondary block to main block. All slots are processed at equal service rate. So, we have to take to into account the total number of arrived slots to obtain accurate service time. All service events may not same time to complete since the number of slot varies. So, the time a slot may spend in main block may vary from one slot to another. We have calculated total service time (total time required for the main block to process all slots) and time when main block has I slots during service event.

TimeService =

Timei\_Service(i) =

where μ = service rate

ℯ= Total number of serviced slots

ṫi= Number of time i-th slot has received service

* **Terms of Events- Arrival of slots to main block from secondary block:**

When main block will process slots, no arrived slots will be queued into the block. Instead, arrived slots will enter into secondary block λ. After main block has processed its blocks, slots from secondary block will be queued into main block. We assume that the process of slots queued into main block from secondary block takes 1 second each time. We calculate the number of slots queued into secondary block during this time.

TimeQueuing(i) =

Where,

ℼ= Count (total i slots have queued into main block)

ℽ= Count (Events when slots queued into main block)

**Probability calculation:**

Applying all these, we can write the probability equation as –

p[i]= \* +

Where,

Arrival time, ℼ =

Service time, T𝝁 =

System time, Tϭ= Arrival\_Time + Service\_Time + Queuing Time

𝜶i= No. of times i-th slot has been queued into main block

= No. of times i-th slot has been serviced

βi= No. of times i number of slots queued from secondary block

γ= No. of times slots queued from secondary to main block

Here, we assume that each slot takes 1s to move from secondary block to primary block

Here, Σp[i] = 1

A figure of our system is attached below

A screenshot of a social media post

Description automatically generated