

# Common Random Numbers (CRN)

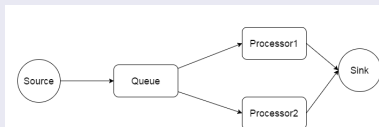
W. M. Song 桑慧敏  
Tsing Hua Univ. 清華大學

May 21, 2016

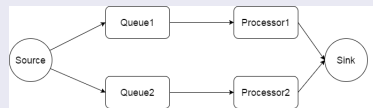
# MM2: One and Two queues

- Goal: Estimate the the diff(Queue-Wait-Time) in two systems with and without CRN and Warmup (Replication: 5, Warmup: 1000)

## System1-MM2 with one queue



## System2-MM2 with two queues



# M/M/2 with One Queue: Analytical

## Notations

- $\lambda, \mu$ : arrival rate, service rate
- $L$ : Length (no. of people wait) in system
- $L_q$ : Length in queue
- $W$ : Wait time in system
- $W_q$ : Wait time in queue
- $\rho = \frac{\lambda}{2\mu}$

## Analytical Solution

- $E(L) = \frac{2\rho}{1-\rho^2}$
- $E(L_q) = \frac{2\rho^3}{1-\rho^2}$
- $E(W) = E(L)/\lambda = \frac{1}{\mu(1-\rho^2)}$
- $E(W_q) = \frac{\rho^2}{\mu(1-\rho^2)}$

# M/M/2 with One Queue: Analytical and Simulation

- The inter-arrival time in Source with: Weibull(0, 2, 1, 0)
- The process time in Processors: Weibull(0, 1, 1, 0)
- Use CRN for inter-arrival times and process times
- Experiment → Run to time: 3500 (Arrivals: 500), Replication: 5

## Analytical

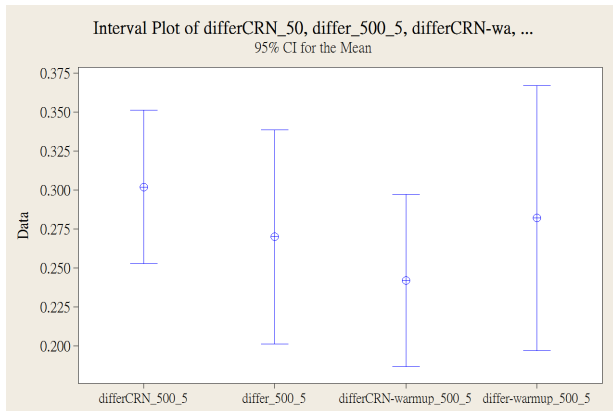
		Mean
1Q	$E(L_q)$	0.033
	$E(W_q)$	0.067

## Simulation

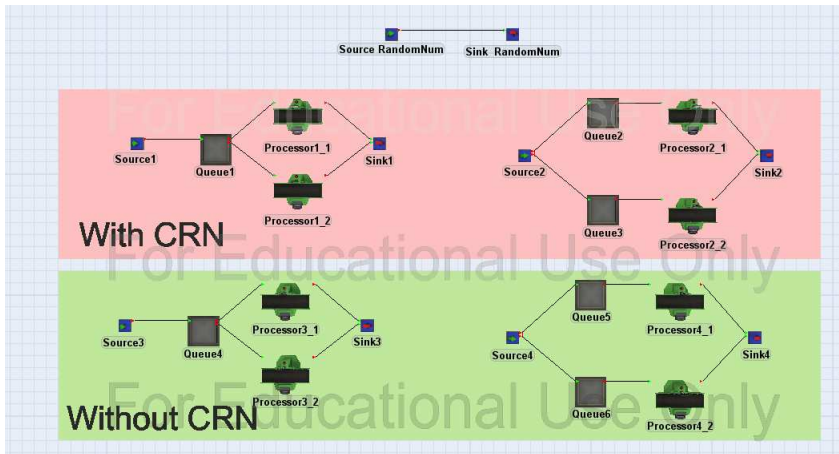
		Mean	SE
1Q	$\hat{E}(L_q)$	0.038	0.0037
	$\hat{E}(W_q)$	0.067	0.0075

# Output Analysis: Interval Estimation

- Goal: Estimate the  $\text{diff}(\text{Queue-Wait-Time})$  in two systems with and without CRN and Warmup (Replication: 5, Warmup: 1000)
- Interval Estimation



# FlexSim Model



# Global Tables

- **InitialValues**: Sample Size, Warmup period
- **RandomNumber**: Record Random Number
- **ArrivalTime**: Record arrival times for each item
- **WQtableCRN**, Record Queue-Wait-Time with CRN
- **WQtable**, Record Queue-Wait-Time without CRN
- **WQ**, Record Queue-Wait-Time with and without CRN and Warmup
- **IndexCheck**, Check whether two systems are completed.
- **StatisticDiffer**, Record the diff(Queue-Wait-Time) in two systems with and without CRN and Warmup

# Create Global Table-Random Number

- Tools → Global Tables → Add
- Table Name: **RandomNumber** , Row: 5000 , Columns: 4 , Clear on Reset
  - Column 1 : Common Random Numbers for Source1, 2, 3
  - Column 2 : Common Random Numbers for Processor1, 2, 3
  - Column 3 : Common Random Numbers for Source4
  - Column 4 : Common Random Numbers for Processor4
- Table Name: **InitialValues** , Row: 3 , Columns: 1
  - Row 1 : Sample Size
  - Row 2 : Start Customer With No Warmup Time
  - Row 3 : Start Customer With Warmup Time

Global Table - RandomNumber

Name: RandomNumber Rows: 5000.00 Columns: 4.00  Clear on Reset

	Random_Source	Random_Processor	Random_Source4	Random_Processor4
Row 2	0.00	0.00	0.00	0.00
Row 3	0.00	0.00	0.00	0.00
Row 4	0.00	0.00	0.00	0.00
Row 5	0.00	0.00	0.00	0.00
Row 6	0.00	0.00	0.00	0.00
Row 7	0.00	0.00	0.00	0.00

Global Table - InitialValues

Name: InitialValues Rows: 3.00 Columns: 1.00  Clear on Reset

	Col 1
SampleSize	2000.00
NoWarmup	0.00
WarmupNum	3000.00



# Create Global Table-Record Arrival Time

- Tools → Global Tables → Add
- Table Name: **ArrivalTime** , Row: 5000 , Columns: 4 , Clear on Reset
  - Column 1 : Arrival Time of Source1
  - Column 2 : Arrival Time of Source2
  - Column 3 : Arrival Time of Source3
  - Column 4 : Arrival Time of Source4

	CRN Sys1-Arrival time	CRN Sys2-Arrival time	Sys1-Arrival time	Sys2-Arrival time
Row 1	0.00	0.00	0.00	0.00
Row 2	0.00	0.00	0.00	0.00
Row 3	0.00	0.00	0.00	0.00
Row 4	0.00	0.00	0.00	0.00
Row 5	0.00	0.00	0.00	0.00
Row 6	0.00	0.00	0.00	0.00

# Create Global Table

- Tools → Global Tables → Add
- Table Name: **WQtableCRN**, Row: 5000, Columns: 7, Clear on Reset
- Table Name: **WQtable**, Row: 5000, Columns: 7, Clear on Reset
  - Column 1: The enter queue time in system 1 (1 Queue)
  - Column 2: The leave queue time in system 1 (1 Queue)
  - Column 3: The wait time in queue in system 1 (1 Queue)
  - Column 4: The enter queue time in system 2 (2 Queue)
  - Column 5: The leave queue time in system 2 (2 Queue)
  - Column 6: The wait time in queue in system 2 (2 Queue)
  - Column 7: The difference of wait time in queue (system 2 - system 1)
- Table Name: **WQ**, Row: 2, Columns: 4, Clear on Reset
  - Row 1: Average WQ With CRN
  - Row 2: Average WQ Without CRN

The screenshot shows three Global Table configuration windows. The first window, 'Global Table - WQtableCRN', shows a table with 5000 rows and 7 columns. The second window, 'Global Table - WQtable', shows a table with 5000 rows and 7 columns. The third window, 'Global Table - WQ', shows a table with 2 rows and 4 columns.

	EnterSys1	LeaveSys1	WQ(Sys1)	EnterSys2	LeaveSys2	WQ(Sys2)	Differ(Sys2-Sys1)
Customer 1	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 2	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	EnterSys1	LeaveSys1	WQ(Sys1)	EnterSys2	LeaveSys2	WQ(Sys2)	Differ(Sys2-Sys1)
	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 2	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Row 3	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	WQ(Sys1)	WQ(Sys2)	WQ(Sys1)warmup	WQ(Sys2)warmup
CRN	0.00	0.00	0.00	0.00
Without CRN	0.00	0.00	0.00	0.00

# Create Global Table-Warmup and Wait Time

- Tools → Global Tables → Add
- Table Name: **IndexCheck**, Row: 2, Columns: 1, Clear on Reset
  - Row 1 : Check 2 Systems With CRN
  - Row 2 : Check 2 Systems Without CRN
- Table Name: **StatisticDiffer**, Row: 2, Columns: 2, Clear on Reset
  - Row 1 : Average Difference of WQ for Each Customer With CRN
  - Row 2 : Average Difference of WQ for Each Customer Without CRN
  - Column 1 : Without Warmup
  - Column 2 : With Warmup

Global Table - IndexCheck

Name: IndexCheck Rows: 2.00 Columns: 1.00  Clear on Reset

CRN	0.00
NO CRN	0.00

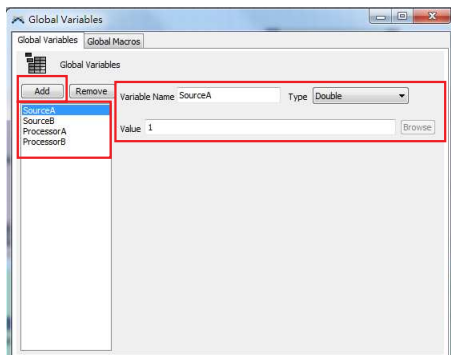
Global Table - StatisticDiffer

Name: StatisticDiffer Rows: 2.00 Columns: 2.00  Clear on Reset

	No Warmup	Warmup
CRN	0.00	0.00
No CRN	0.00	0.00

# Create Global Variables

- Tools → Global Variables → Add
  - Variable Name: `SourceA` , Type: Double , Value=1
  - Variable Name: `SourceB` , Type: Double , Value=2
  - Variable Name: `ProcessorA` , Type: Double , Value=1
  - Variable Name: `ProcessorB` , Type: Double , Value=1



# Create Random Numbers

- Source RandomNum → Source → Inter-Arrivaltime
- Source RandomNum → Triggers → OnExit

The image shows two windows from a simulation software. The left window is titled "Source RandomNum Properties" and has several tabs: "Source", "Flow", "Triggers", "Labels", "General", and "Statistics". The "Source" tab is selected. In this tab, the "Arrival Style" is set to "Inter-Arrival Time", the "FlowItem Class" is "Box", and the "Inter-Arrivaltime" is set to 0. The right window is titled "/Source RandomNum - OnExit" and contains a block of C++ code. The code defines a custom code block for the "OnExit" trigger, which generates random numbers for the "RandomNumber" table using the uniform distribution function. The code also checks for initial values and stops output if the initial value is reached.

```
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 settablenum("RandomNumber",getoutput(current)+1,1,uniform(0,1));
7 settablenum("RandomNumber",getoutput(current)+1,2,uniform(0,1));
8 settablenum("RandomNumber",getoutput(current)+1,3,uniform(0,1));
9 settablenum("RandomNumber",getoutput(current)+1,4,uniform(0,1));
10
11 if(getoutput(current)==gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1)-1)
12 {
13     stopoutput(current);
14 }
15 }
16
```

# Start System1 and System2

- Sink RandomNum → Triggers → OnEntry

```
~/Sink RandomNum - OnEntry  
Sink RandomNum - OnEntry  
1 /**Custom Code*/  
2 treenode item = parnode(1);  
3 treenode current = ownerobject(c);  
4 int port = parval(2);  
5  
6  
7 if(getinput(current)==gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1))  
8 {  
9     openoutput(node("/Source1",model()));  
10    openoutput(node("/Source2",model()));  
11    openoutput(node("/Source3",model()));  
12    openoutput(node("/Source4",model()));  
13 }
```

# Inter-Arrivaltime of Source

## With CRN

- Source1, Source2 → Source → Inter-Arrivaltime

```
Source1 - Inter-Arrival Time
1 /**Custom Code*/
2 treenode current = ownerobject(c);
3
4 return -(SourceB*pow((log(1-gettablenum("RandomNumber", getoutput(current)+1,1))), (1/SourceA)));

Source2 - Inter-Arrival Time
1 /**Custom Code*/
2 treenode current = ownerobject(c);
3
4 return -(SourceB*pow((log(1-gettablenum("RandomNumber", getoutput(current)+1,1))), (1/SourceA)));
```

# Inter-Arrivaltime of Source

## Without CRN

- Source3, Source4 → Source → Inter-Arrivaltime

The image shows two overlapping NetLogo code windows. The top window is titled "/Source3 - Inter-Arrival Time" and contains the following code:

```
1 /**Custom Code*/  
2 treenode current = ownerobject(c);  
3  
4 return -(SourceB*pow((log(1-gettablenum("RandomNumber",getoutput(current)+1,1))), (1/SourceA)));
```

The bottom window is titled "/Source4 - Inter-Arrival Time" and contains the following code:

```
1 /**Custom Code*/  
2 treenode current = ownerobject(c);  
3  
4 return -(SourceB*pow((log(1-gettablenum("RandomNumber",getoutput(current)+1,3))), (1/SourceA)));
```

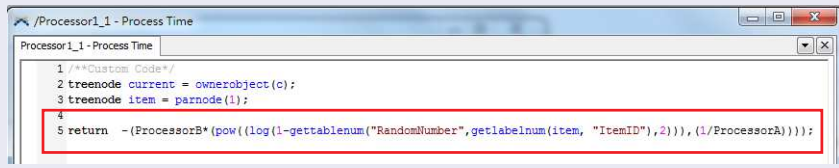
In both code snippets, the numbers 1 and 3 in the exponent of the power function are highlighted with yellow boxes. The entire code blocks in both windows are enclosed in red rectangular boxes.



# Process Time of Processor

## With CRN

- Processor1\_1, Processor1\_2 → Processor → Process Time
- Processor2\_1, Processor2\_2 → Processor → Process Time



```
1 /**Custom Code*/  
2 treenode current = ownerobject(c);  
3 treenode item = parnode(1);  
4  
5 return -(ProcessorB*(pow((log(1-gettablenum("RandomNumber", getlabelnum(item, "ItemID"), 2))), (1/ProcessorA))));
```

# Process Time of Processor

## Without CRN

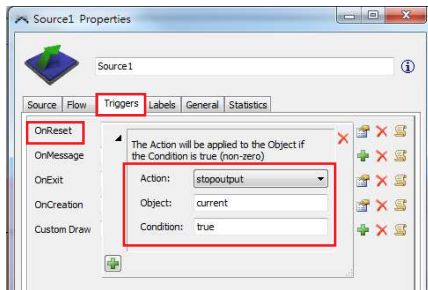
- Processor3\_1, Processor3\_2 → Processor → Process Time
- Processor4\_1, Processor4\_2 → Processor → Process Time

```
Processor3_1 - Process Time
Processor3_1 - Process Time
1 /**Custom Code*/
2 treenode current = ownerobject(c);
3 treenode item = parnode(1);
4
5 return -(ProcessorB*(pow((log(1-gettablenum("RandomNumber",getlabelnum(item, "ItemID"),2)), (1/ProcessorA))));

Processor4_1 - Process Time
Processor4_1 - Process Time
1 /**Custom Code*/
2 treenode current = ownerobject(c);
3 treenode item = parnode(1);
4
5 return -(ProcessorB*(pow((log(1-gettablenum("RandomNumber",getlabelnum(item, "ItemID"),2)), (1/ProcessorA))));
```

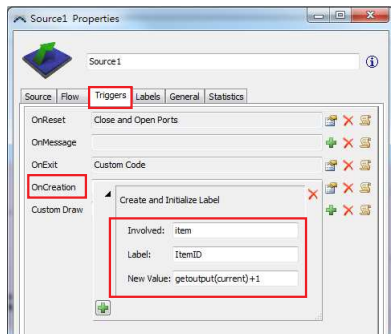
# Close Output of Source

- Source1 → Triggers → OnReset → Close and Open Ports
- Source2 → Triggers → OnReset → Close and Open Ports
- Source3 → Triggers → OnReset → Close and Open Ports
- Source4 → Triggers → OnReset → Close and Open Ports



# Create Labels

- Source1 → Triggers → OnCreation → Create and Initialize Label
- Source2 → Triggers → OnCreation → Create and Initialize Label
- Source3 → Triggers → OnCreation → Create and Initialize Label
- Source4 → Triggers → OnCreation → Create and Initialize Label



# Record Arrival Time

## With CRN

- Source1 → Triggers → OnExit
- Source2 → Triggers → OnExit

The image shows two overlapping NetLogo source code windows. The top window is titled "/Source1 - OnExit" and the bottom window is titled "/Source2 - OnExit". Both windows contain custom code for recording arrival times. In the top window, line 6 is highlighted with a red box and contains the code: `6 settablenum("ArrivalTime", getlabelnum(item, "ItemID"), 1, time());`. In the bottom window, line 6 is also highlighted with a red box and contains the code: `6 settablenum("ArrivalTime", getlabelnum(item, "ItemID"), 2, time());`. The code in both windows includes comments and other logic for handling items and outputting current values.

```
Source1 - OnExit
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 settablenum("ArrivalTime", getlabelnum(item, "ItemID"), 1, time());
7
8 if(getlabelnum(item, "ItemID")==gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1))
9 {
10   stopoutput(current);
11 }
12

Source2 - OnExit
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int rownumber = parval(2);
5
6 settablenum("ArrivalTime", getlabelnum(item, "ItemID"), 2, time());
7
8 if(getlabelnum(item, "ItemID")==gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1))
9 {
10   stopoutput(current);
11 }
12
```

# Record Arrival Time

## Without CRN

- Source1 → Triggers → OnExit
- Source2 → Triggers → OnExit

```
Source3 - OnExit
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 settablenum("ArrivalTime", getlabelnum(item, "ItemID"), 3, time());
7
8 if(getlabelnum(item, "ItemID")==gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1))
9 {
10   stopoutput(current);
11 }
12

Source4 - OnExit
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int rownumber = parval(2);
5
6 settablenum("ArrivalTime", getlabelnum(item, "ItemID"), 4, time());
7
8 if(getlabelnum(item, "ItemID")==gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1))
9 {
10   stopoutput(current);
11 }
12
```

# Set Enter and Leave Time-System 1

- Queue1, Queue4 → Triggers → OnEntry
- Queue1, Queue4 → Triggers → OnExit
- Record Enter Time, Leave Time in Global Table

## With CRN

```
~/Queue1 - OnEntry
Queue1 - OnEntry
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 settablenum("WQtableCRN",getlabelnum(item,"ItemID"),1,time());

~/Queue1 - OnExit
Queue1 - OnExit
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 settablenum("WQtableCRN",getlabelnum(item,"ItemID"),2,time());
```

## Without CRN

```
~/Queue4 - OnEntry
Queue4 - OnEntry
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 settablenum("WQtable",getlabelnum(item,"ItemID"),1,time());

~/Queue4 - OnExit
Queue4 - OnExit
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 settablenum("WQtable",getlabelnum(item,"ItemID"),2,time());
```

# Set Enter and Leave Time-System2

- Queue2, Queue3, Queue5, Queue6 → Triggers → OnEntry
- Queue2, Queue3, Queue5, Queue6 → Triggers → OnExit

## With CRN

The image shows two code windows for Queue2. The top window, titled 'Queue2 - OnEntry', contains the following code: 

```
1 /**Custom Code*/  
2 treenode item = parnode(1);  
3 treenode current = ownerobject(c);  
4 int port = parval(2);  
6 settablenum("WQtableCRN", getlabelnum(item, "ItemID"), 4, time());
```

 The number '4' in the last line is highlighted in a yellow box. The bottom window, titled 'Queue2 - OnExit', contains the following code: 

```
1 /**Custom Code*/  
2 treenode item = parnode(1);  
3 treenode current = ownerobject(c);  
4 int port = parval(2);  
6 settablenum("WQtableCRN", getlabelnum(item, "ItemID"), 5, time());
```

 The number '5' in the last line is highlighted in a yellow box. Red boxes highlight the code lines in both windows.

## Without CRN

The image shows two code windows for Queue5. The top window, titled 'Queue5 - OnEntry', contains the following code: 

```
1 /**Custom Code*/  
2 treenode item = parnode(1);  
3 treenode current = ownerobject(c);  
4 int port = parval(2);  
6 settablenum("WQtable", getlabelnum(item, "ItemID"), 4, time());
```

 The number '4' in the last line is highlighted in a yellow box. The bottom window, titled 'Queue5 - OnExit', contains the following code: 

```
1 /**Custom Code*/  
2 treenode item = parnode(1);  
3 treenode current = ownerobject(c);  
4 int port = parval(2);  
6 settablenum("WQtable", getlabelnum(item, "ItemID"), 5, time());
```

 The number '5' in the last line is highlighted in a yellow box. Red boxes highlight the code lines in both windows.



# Wait Time In Queue; With CRN

- Sink1 → Triggers → OnEntry

```
Sink1 - OnEntry
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 double Sum=0;
7 double SumW=0;
8 double SumWQ1=0;
9 double SumWQ2=0;
10 if (getinput(current) == gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1))
11 {
12     settablenum("IndexCheck",1,1,gettablenum("IndexCheck",1,1)+1);
13     //calculate WQ
14     for(int k=1 ;k<=gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1);k++)
15     {
16         settablenum("WQtableCRN",k,3,gettablenum("WQtableCRN",k,2)-gettablenum("WQtableCRN",k,1));
17     }
18     //calculate average WQ
19     for(int k=gettablenum("InitialValues",2,1)+1 ;k<=gettablenum("InitialValues",1,1)+gettablenum("InitialValues",2,1);k++)
20     {
21         SumWQ1 = SumWQ1 + gettablenum("WQtableCRN",k,3);
22     }
23     settablenum("WQ",1,1,SumWQ1/gettablenum("InitialValues",1,1));
24     for(int k=gettablenum("InitialValues",3,1)+1 ;k<=gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1);k++)
25     {
26         SumWQ2 = SumWQ2 + gettablenum("WQtableCRN",k,3);
27     }
28     settablenum("WQ",1,3,SumWQ2/gettablenum("InitialValues",1,1));
29
30     if(gettablenum("IndexCheck",1,1)==2)
31     {
32         //calculate differ
33         for(int k=gettablenum("InitialValues",2,1)+1 ;k<=gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1);k++)
34         {
35             settablenum("WQtableCRN",k,7,gettablenum("WQtableCRN",k,6)-gettablenum("WQtableCRN",k,3));
36         }
37         for(int k=gettablenum("InitialValues",2,1)+1 ;k<=gettablenum("InitialValues",1,1);k++)
38         {
39             Sum = Sum + gettablenum("WQtableCRN",k,7);
40         }
41     }
42 }
```

# Wait Time In Queue; With CRN

- Sink2 → Triggers → OnEntry

```
Sink2 - OnEntry
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 double Sum=0;
7 double SumW=0;
8 double SumWQ1=0;
9 double SumWQ2=0;
10 if (getinput(current) == gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1))
11 {
12     settablenum("IndexCheck",1,1,gettablenum("IndexCheck",1,1)+1);
13     //calculate WQ
14     for(int k=1 ;k<=gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1);k++)
15     {
16         settablenum("WQtableCRN",k,6,gettablenum("WQtableCRN",k,5)-gettablenum("WQtableCRN",k,4));
17     }
18     //calculate average WQ
19     for(int k=gettablenum("InitialValues",2,1)+1 ;k<=gettablenum("InitialValues",1,1)+gettablenum("InitialValues",2,1);k++)
20     {
21         SumWQ1 = SumWQ1 + gettablenum("WQtableCRN",k,6);
22     }
23     settablenum("WQ",1,2,SumWQ1/gettablenum("InitialValues",1,1));
24     for(int k=gettablenum("InitialValues",3,1)+1 ;k<=gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1);k++)
25     {
26         SumWQ2 = SumWQ2 + gettablenum("WQtableCRN",k,6);
27     }
28     settablenum("WQ",1,4,SumWQ2/gettablenum("InitialValues",1,1));
29
30     if(gettablenum("IndexCheck",1,1)==2)
31     {
32         //calculate differ
33         for(int k=1 ;k<=gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1);k++)
34         {
35             settablenum("WQtableCRN",k,7,gettablenum("WQtableCRN",k,6)-gettablenum("WQtableCRN",k,3));
36         }
37         for(int k=gettablenum("InitialValues",2,1)+1 ;k<=gettablenum("InitialValues",1,1);k++)
38         {
39             Sum = Sum + gettablenum("WQtableCRN",k,7);
40         }
41     }
42 }
```

# Wait Time In Queue; Without CRN

- Sink3 → Triggers → OnEntry

```
Sink3 - OnEntry
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 double Sum=0;
7 double SumW=0;
8 double SumWQ1=0;
9 double SumWQ2=0;
10 if (getinput(current) == gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1))
11 {
12     settablenum("IndexCheck",2,1,gettablenum("IndexCheck",2,1)+1);
13     //calculate WQ
14     for(int k=1 ;k<=gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1);k++)
15     {
16         settablenum("WQtable",k,3,gettablenum("WQtable",k,2)-gettablenum("WQtable",k,1));
17     }
18     //calculate average WQ
19     for(int k=gettablenum("InitialValues",2,1)+1 ;k<=gettablenum("InitialValues",1,1)+gettablenum("InitialValues",2,1);k++)
20     {
21         SumWQ1 = SumWQ1 + gettablenum("WQtable",k,3);
22     }
23     settablenum("WQ",2,1,SumWQ1/gettablenum("InitialValues",1,1));
24     for(int k=gettablenum("InitialValues",3,1)+1 ;k<=gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1);k++)
25     {
26         SumWQ2 = SumWQ2 + gettablenum("WQtable",k,3);
27     }
28     settablenum("WQ",2,3,SumWQ2/gettablenum("InitialValues",1,1));
29
30     if(gettablenum("IndexCheck",2,1)==2)
31     {
32         //calculate differ
33         for(int k=1 ;k<=gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1);k++)
34         {
35             settablenum("WQtable",k,7,gettablenum("WQtable",k,6)-gettablenum("WQtable",k,3));
36         }
37         for(int k=gettablenum("InitialValues",2,1)+1 ;k<=gettablenum("InitialValues",1,1);k++)
38         {
39             Sum = Sum + gettablenum("WQtable",k,7);
40         }
41     }
42 }
```

# Wait Time In Queue: Without CRN

- Sink4 → Triggers → OnEntry

Sink4 - OnEntry

```
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 double Sum=0;
7 double SumW=0;
8 double SumWQ1=0;
9 double SumWQ2=0;
10 if (getinput(current) == gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1))
11 {
12     settablenum("IndexCheck",2,1,gettablenum("IndexCheck",2,1)+1);
13     //calculate WQ
14     for(int k=1 ;k<=gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1);k++)
15     {
16         settablenum("WQtable",k,6,gettablenum("WQtable",k,5)-gettablenum("WQtable",k,4));
17     }
18     //calculate average WQ
19     for(int k=gettablenum("InitialValues",2,1)+1 ;k<=gettablenum("InitialValues",1,1)+gettablenum("InitialValues",2,1);k++)
20     {
21         SumWQ1 = SumWQ1 + gettablenum("WQtable",k,6);
22     }
23     settablenum("WQ",2,2,SumWQ1/gettablenum("InitialValues",1,1));
24     for(int k=gettablenum("InitialValues",3,1)+1 ;k<=gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1);k++)
25     {
26         SumWQ2 = SumWQ2 + gettablenum("WQtable",k,6);
27     }
28     settablenum("WQ",2,4,SumWQ2/gettablenum("InitialValues",1,1));
29
30     if (gettablenum("IndexCheck",2,1)==2)
31     {
32         //calculate differ
33         for(int k=1 ;k<=gettablenum("InitialValues",1,1)+gettablenum("InitialValues",3,1);k++)
34         {
35             settablenum("WQtable",k,7,gettablenum("WQtable",k,6)-gettablenum("WQtable",k,3));
36         }
37     }
38 }
```