

PAPER • OPEN ACCESS

The simulation of three crossroad traffic queueing systems using petri nets and colored petri nets program

To cite this article: A Wicaksono and M A Rudhito 2019 *J. Phys.: Conf. Ser.* **1307** 012012

View the [article online](#) for updates and enhancements.



IOP | ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the [collection](#) - download the first chapter of every title for free.

The simulation of three crossroad traffic queueing systems using petri nets and colored petri nets program

A Wicaksono^{1*}, M A Rudhito²

¹Student of Master of Mathematics Education Study Program, Sanata Dharma University, Yogyakarta, Indonesia

² Master of Mathematics Education Study Program, Sanata Dharma University, Yogyakarta, Indonesia

*anungwicaksono10@gmail.com

Abstract. This research aims to develop a simulation of traffic flow at the crossroads, that's Monjali, Kentungan and Gejayan crossroads in Yogyakarta with and without underpass. The simulation using petri nets and colored petri nets programs. We assume that each crossroad has the same density, and we look that the traffic flow just went one way. The results of the study show that the queueing system for three traffic intersections can be modeled using petri nets. Furthermore, a queue simulation can be carried out with the help of a colored petri nets program. Simulation results indicate a buildup at certain junctions without underpasses and with underpasses

1. Introduction

The construction of an underpass on the kentungan crossroad located on the northern ring road is intended to facilitate the flow of traffic around the crossroads. On the north ring road itself, this crossroad make a system with another crossroads which is monjali crossroads and gejayan crossroads. The underpass is built for the direct traffic in the middle of the road can go through without stopped by the traffic light on the crossroads. For the situation at the intersection itself, it looks like it will be smoother, but justify the traffic flow will be smoother if the traffic system involves the nearest crossroad, namely the crossroad of monjali and the crossroad of gejayan. It can be a smooth current at the kentungan crossroad, but the system with another crossroad will be the same or there will be smooth. Then a simulation will be made using the colored petri net program when without using an underpass and when using an underpass. The purpose of this research is to develop a simulation of traffic flow at the crossroads, that's Monjali, Kentungan and Gejayan crossroad in Yogyakarta with and without underpass with colored petri nets program and using token on petri net program we can estimate the build-up happen on each crossroad.

2. Basic theory

2.1. Petri net

According to [1,2] Petri net was first developed by C.A. Petri in the early 1960s. Petri net is a tool for modeling discrete event systems other than using previously known automata. Each automata can be changed to Petri net. On Petri net "events" are related to transitions. For an "event" to occur,



some conditions must be fulfilled first. Information about events and circumstances each of these is expressed by transition and place. Place can function as input or output of a transition. Place as input states the condition that must be fulfilled so that transitions can occur. After the transition occurs, the situation will change. Place which states that this condition is the output of the transition.

Definition 4.0.1 Petri net is 4-tuple (P, T, A, W) with

- P: finite set of places, $P = \{P_1, P_2, \dots, P_n\}$.
- T: finite set of transitions, $T = \{T_1, T_2, \dots, T_n\}$.
- A: arc set, $A \subseteq (P \times T) \cup (T \times P)$,
- w: weight function, $W = A \rightarrow \{1, 2, 3, \dots\}$

2.2. Colored petri net

Based on [3,4] Colored Petri Nets (CP-nets or CPNs) is a graphical language for constructing models of concurrent systems and analyzing their properties. CP-nets is a discrete-event modeling language combining the capabilities of Petri nets with the capabilities of a high-level programming language. Petri nets provide the foundation of the graphical notation and the basic primitives for modeling concurrency, communication, and synchronization. A well-known program for working with CPNs is CPN Tools.

Definition 2.1. A net is a tuple $N = (P, T, A, \Sigma, C, N, E, G, I)$, where:

- P is a set of places.
- T is a set of transitions.
- A is a set of arcs

In CPNs sets of places, transitions and arcs are pairwise disjoint $P \cap T = P \cap A = T \cap A = \emptyset$

- Σ is a set of color sets defined within CPN model. This set contains all possible colors, operations, and functions used within CPN.
- C is a color function. It maps places in P into colors in Σ .
- N is a node function. It maps A into $(P \times T) \cup (T \times P)$.
- E is an arc expression function. It maps each arc $a \in A$ into the expression E. The input and output types of the arc expressions must correspond to the type of the nodes the arc is connected to.

The use of node function and arc expression function allows multiple arcs to connect the same pair of nodes with different arc expressions.

- G is a guard function. It maps each transition $t \in T$ to a guard expression g. The output of the guard expression should evaluate to Boolean value: true or false.
- I is an initialization function. It maps each place p into an initialization expression i. The initialization expression must evaluate to a multiset of tokens with a color corresponding to the color of the place C(p).

3. Discussion

This discussion will be explained the petri net and simulation in three crossroads. According to [5] the model of single crossroad had already been modelled but this research talk about CPN and use it to get the simulation of traffic flow between three crossroads. Although it's already been discussed on [6], none have talked about simulation between three crossroads. The model of single crossroad had already been discussed in [7], on this part we look the other point of view from crossroads. We can take in every crossroads turn left can pass directly without stopping when the light turns red, while the other two cannot and need to wait until the light turns into green. So this case is only seen the position of the vehicle movement that is straight and turns right.

Figure 1 explains that every crossroad have the same position, there are from north (N) can go to south(S) and west(W), from east(E) can go to west(W) and north(N), from south(S) can go to north(N) and east(E), and from west(W) can go to east(E) and south(S). Then we can see the three crossroads become like figure 2.

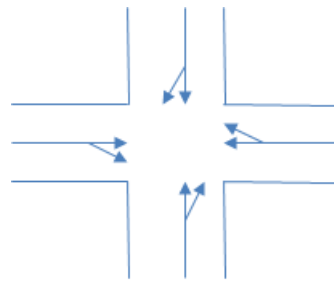


Figure 1. Single crossroad.

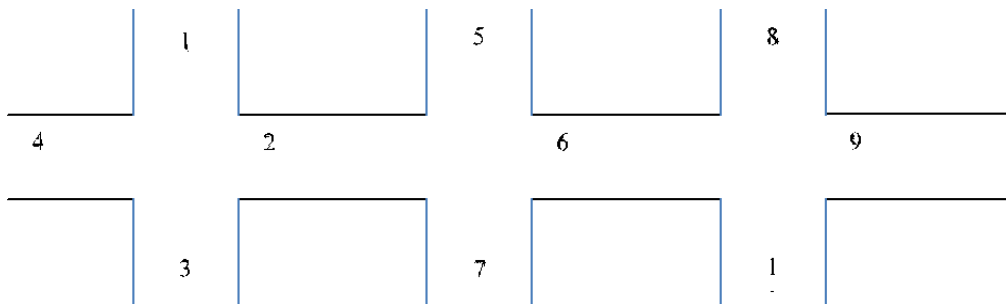


Figure 2. Three crossroads.

By following the pattern so we can make some direction, from 1 can go to 3 and 4, from 2 can go to 4 and 1, from 3 can go to 1 and 2, and from 4 can go to 2 and 3. The second crossroad, from 5 can go to 7 and 2, from 6 can go to 2 and 5, from 7 can go to 5 and 6, and from 2 can go to 6 and 7. The third crossroad, from 8 can go to 10 and 6, from 9 can go to 6 and 8, from 10 can go to 8 and 9, and from 6 can go to 9 and 10. Thus in petri net we can write $P = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}\}$ as position 1 to 10. And for transition we can write $T = \{T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, T_{10}\}$ as transition when green light on. We have the assumption:

1. The direction of the vehicle observed just straightway, because it is assumed that the vehicle that exits from position 4 to position 2 will be filled with vehicles entering from position 1 and position 3 as well as 2 other crossroads
2. Vehicles that pass from the last crossroads are assumed to be replenished at the first crossroad according to a closed queueing system [8].

Petri net model can be drawn as in figure 3.

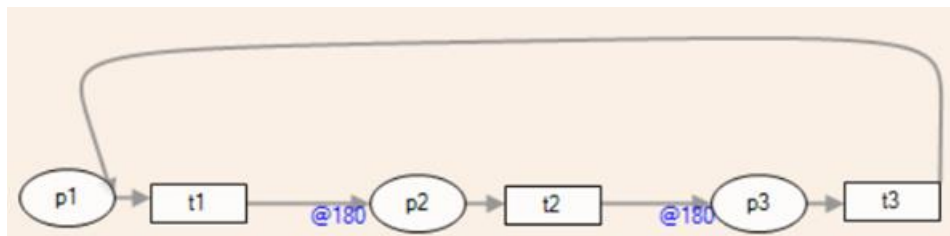


Figure 3. Model of petri net for three crossroads with assumption.

From figure 3 we found that the time between each crossroad to another crossroad was 180 seconds. We can make 3x3 matrix from that petri net as $A = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & 0 & -1 \end{bmatrix}$. On this model we know that each traffic light can be list on the Table 1 and then can be inputted on CPN Tools application as in figure 4.

Table 1. Traffic light within three crossroads.

Crossroad Name	Red Light	Green Light
Monjali	145 in place 1	38 in transition 1
Kentungan	140 in place 2	63 in transition 2
Gejayan	133 in place 3	53 in transition 3

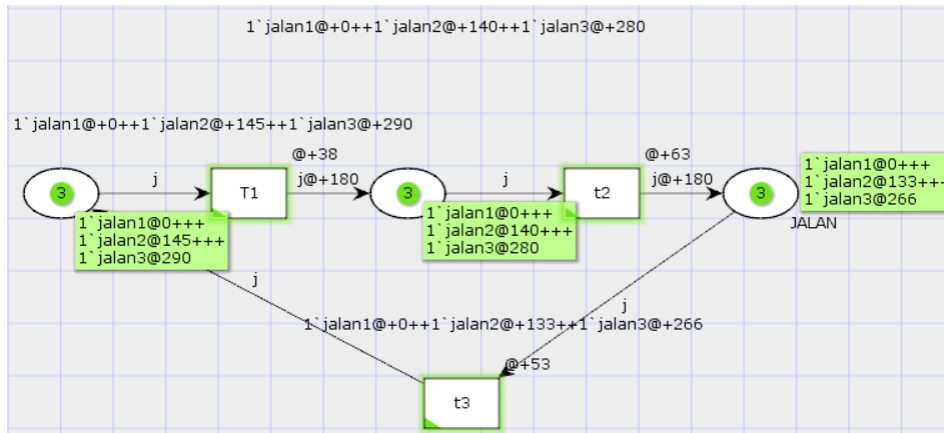


Figure 4. Three crossroads without underpass.

Every crossroads got 3 tokens (density of vehicle) and transition is green light, we assume that when at begin each token on each crossroads move to another “place”, after then followed by another token move after time of red light begin in the first crossroad is 145 seconds, second crossroad is 140 seconds and the last 133 seconds and so on. After repeat 30 times that simulation shows that there is build up on specific crossroads, it can be Monjali crossroad, or another two, the model on CPN Tools can be shown as figure 5. From figure 5 we get 3 tokens as the density of vehicle if the initial density is very dense so the time to break the dense of density is very high on specific crossroads.

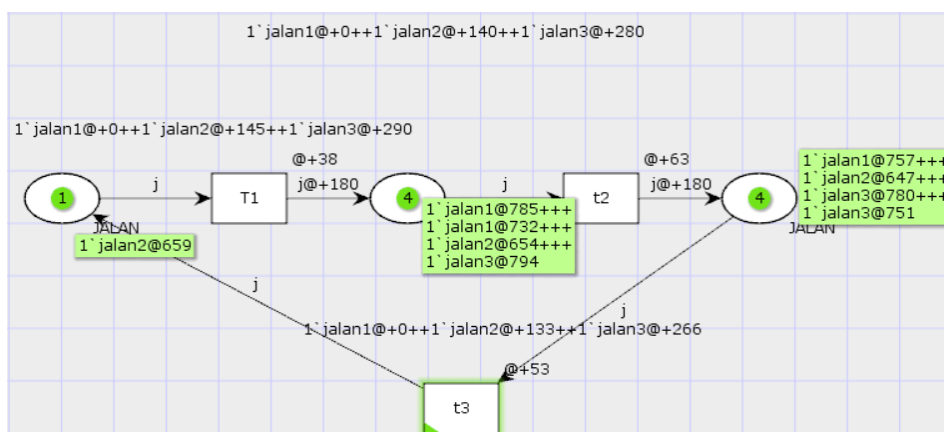


Figure 5. Three crossroads without underpass after 30 times.

On the next condition, we have underpass in kentungan crossroad then the time needed to go to gejayan crossroad will become 360 seconds so the model will shown in figure 6. With the same application, we route 20 times so the model as in figure 7.

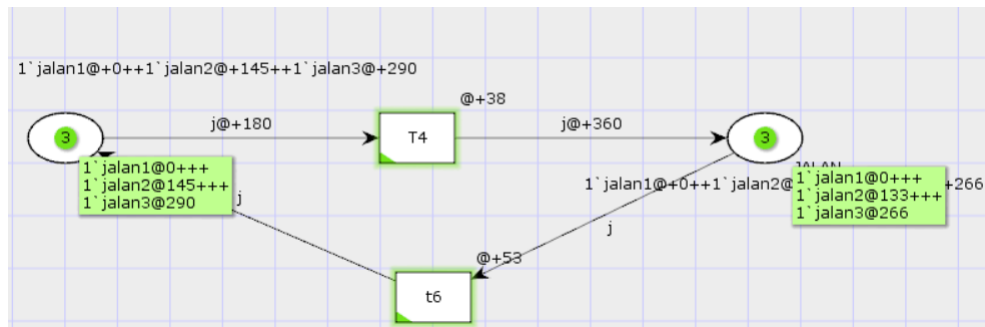


Figure 6. Three crossroads with underpass.

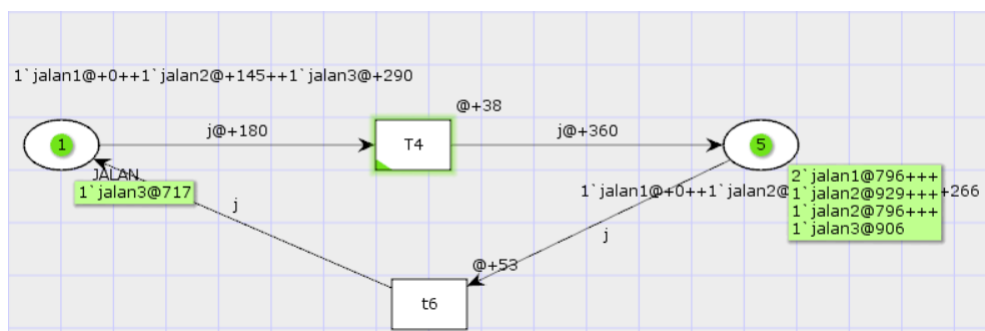


Figure 7. Three crossroads without underpass after 20 times.

We get another build up on the last crossroad, with underpass on kentungan crossroads can make another significant change on the last crossroads, because without underpass the gejalan crossroad (last crossroad) get the same effect on a traffic jam, then when we built underpass on kentungan the condition didn't change. In the same way, we can create a table if the token we have starts from 1 to 5 with an example of 1 token representing the state of the vehicle queue so that we can see at the position of each intersection when 30 movements have occurred for the simulation type without underpasses and 20 for the type of simulation with underpass as in table 2.

From table 2, we get that the build-up happened on crossroads without underpass when the token reaches 3. In crossroads, with an underpass, the happened when token reach 2. Although we can see with the steps happened on the loop after that we will try to look with the times that happen on the loop we take 500 seconds and 1000 seconds as the test on CPN Tools application, see table 3.

Table 2. CPN program with variation token.

Token(s)	Token on each crossroads					
	without underpass (30x steps)			with underpass (20x steps)		
	P1	P2	P3	P1	P3	
1	1	1	1	1	1	
2	2	2	2	0	2	
3	1	4	4	1	5	
4	2	5	5	0	8	
5	3	6	6	1	9	

Table 3. CPN with times 500 seconds.

Token(s)	Token on each crossroads				
	without underpass (500 seconds)			with underpass (500 seconds)	
	P1	P2	P3	P1	P3
1	1	0	2	1	1
2	1	2	3	1	3
3	1	4	4	1	5
4	1	6	5	1	7
5	2	7	6	1	9

From the table 3 we can see when we had condition with times as 500 seconds the positions of each token will stack on P2 and P3 with 3 or more token that happened on crossroads without underpass built on P2. with underpass built on P2 we can see the token still stack on P3 with data can be seen on the table 3. After that we try to increase the times to be 1000 seconds and the results are reported in table 4.

Table 4. Data of CPN with 1000 seconds.

Token(s)	Token on each crossroads				
	without underpass (1000 seconds)			with underpass (1000 seconds)	
	P1	P2	P3	P1	P3
1	1	0	2	1	1
2	2	2	2	1	3
3	1	4	4	1	5
4	1	6	5	1	7
5	1	7	7	1	9

Same with the condition at 500 seconds as test the position of each token still manages to stack on P2 and P3 in crossroads without underpass built on P2. With underpass built on P2, the token still stacks on P3. We can take some notice here, even we build some underpass on one crossroad it will affect the other crossroads as long as the crossroads manage the conditions.

4. Conclusion

From the explanation of table 4 and by doing some simulations with CPN we get that for more than 2 tokens at each crossroads without underpasses we get that there is a build-up of the number of tokens which represent the number of vehicles in that position, the build-up usually occurs at kentungan crossroad and gejayan crossroad. while for crossroads with underpasses for tokens more than 2 we always get a token stack at gejayan crossroad. Thus it can be said that the underpass at the kentungan crossroad can only break down the kentungan crossroad itself but create a traffic effect at the gejayan intersection

References

- [1] Subiono 2015 *Aljabar Min-Max Plus dan Terapannya ver 3.0.0* (Surabaya: Institut Teknologi Sepuluh November)
- [2] Tristono T, Cahyono S, Daru S and Utomo P 2018 *Proc. The Third Int. Conf. on Sustainable Infrastructure and Built Environment* vol 147 p 02005

- [3] Jensen K and Kristensen L 2009 *Colored Petri Nets Modeling and Validation of Concurrent Systems* (Berlin: Heidelberg)
- [4] An Y, Wu N, Zhao X, Li X and Chen P 2018 *Appl. Sci.* **8** 141.
- [5] Tolba C, Thomas P, Elmoudni A and Lefebvre D 2003 *Proc. Conf.: Emerging Technologies and Factory Automation* vol 2
- [6] Zhou F, Jing X and Wang M 2013 *J. Comput. Inform. Syst.* **9** 1217
- [7] Benyamin I 2017 *Simulasi dan Pemodelan Lalu Lintas Pada Simpang Bersinyal Untuk Mengurangi Kerugian Antrian* (Surabaya: Institut Teknologi Sepuluh November)
- [8] Rudhito M 2016 *Aljabar Max Plus dan Penerapannya* (Yogyakarta: Universitas Sanata Dharma)
- [9] Wang C 2013 *Urban Transportation Networks: Analytical Modeling of Spatial Dependencies and Calibration Techniques for Stochastic Traffic Simulator* (Massachusetts: Massachusetts Institute of Technology)
- [10] Febbraro A, Giglio D, Sacco N 2015 *IEEE Trans. Intelligent Transportation Systems* **17** 510