**Proceedings of the 8th International**
**IEEE Conference on Intelligent Transportation Systems**
**Vienna, Austria, September 13-16, 2005**

# Traffic Signal Priority/Preemption Control with Colored Petri Nets

Lefei Li, Wei-Hua Lin, and Hongchao Liu

***Abstract*—A Colored Petri Net (CPN) model for traffic signal priority/preemption is described. The model incorporates the priority/preemption request of transit vehicles into traffic signal control. Compared with 'uncolored' Petri Nets (PN), CPN provides a broader modeling capability for traffic signal priority control. A simple priority evaluation procedure is proposed to provide different priorities to multiple priority requests. An example for the queue jump phase modeled with CPN is presented.**

## I. INTRODUCTION

TRANSIT Signal Priority (TSP) control is an important component of Intelligent Transportation Systems. It is aimed at reducing delay for transit vehicles at signalized intersections by temporarily adjusting the traffic signal timing to benefit transit vehicles. A significant difference between priority and preemption is that priority control is much conditional and restrictive while preemption control is not "negotiable" and in most cases subject to only pedestrian time. The "abrupt" and "absolute" nature of preemption makes it a lot easier to operate and most traffic signal controllers have standard preemption functions. The priority control, however, is much flexible.

Petri Nets (PN) is a formal and graphical appealing language known for its effectiveness in modeling discrete event systems. PN was first used to model traffic signal control in 1994 [1]. Recent research in this area focused primarily on implementing signal control logic [2], developing functioning rules for controllers [3], and extending to Hybrid Petri Nets (HPN) to model the urban traffic control structure [4].

Colored Petri Nets (CPN), introduced by Kurt Jensen [5], is a modeling language that is powerful in handling systems in which communication, synchronization, and resource sharing play an important role. CPN has been actively studied and applied to various areas of computer science and technology. CPN models for traffic signals were proposed recently in [6] in which Design/CPN tools are used for design and animation of the CPN models.

In this paper, a CPN model for traffic signal priority/preemption control is proposed. Sec. II provides an overview of the traffic signal priority/preemption control problem with the emphasis on multiple-request priority control. A simple expression for priority evaluation is proposed. Sec. III addresses the basic concepts of CPN and their applications in signal priority control are discussed. The framework of a CPN model is proposed. Sec. IV gives an example showing the queue jump phase modeled with CPN. A discussion of future work is given in Sec. V.

## II. TRAFFIC SIGNAL PRIORITY/PREEMPTION CONTROL

The objective of traffic signal priority/ preemption control is to provide transit vehicles with higher priority over other vehicles in a traffic stream. Real-time calculation and decision-making are critical for traffic signal priority/ preemption control, since priority control is essentially to embed costs into priority rules for decision-making.

A variety of treatments were developed and implemented to adjust normal signal operation in favor of transit and emergency vehicles, which include but not limited to: early green, green extension, queue-jump phase, phase recall, phase suppression, and phase rotation, wherein the early green, green extension, and queue-jump phase are three most often used strategies in the field [7]. Early green strategy starts the priority phase earlier by shortening the green time of the preceding phases without violating the minimum green time and pedestrian time. Green extension strategy is similar to early green strategy in the sense that the green interval is enlarged by shortening following phases so as to keep signal cycle as constant as possible. The strategy of queue-jump phase is more aggressive which allows buses to pull ahead of queue by using other lanes (most often right turn lanes) followed by a short green interval to accommodate their passage. In [8], an optimal timing plan is generated based on an evaluation of a set of candidate signal plans. The evaluation compares total delay under different plans, taking into consideration the delay to all the vehicles in queues but assigning transit vehicles a larger weight so that the signal plan is in favor of transit vehicles.

The concern that TSP may mess up otherwise well coordinated signals and induce excessive delays to non-transit vehicles has long been the barrier for deployment of TSP in North America. Recent advances in Intelligent Transportation Systems (ITS), bus technologies, traffic engineering and other innovations have made it possible to track bus movement and monitor vehicle queuing status at a rather convincible level. How to take advantage of the advancement and incorporate the information into the design process of signal priority algorithms presents a challenge to researchers and practitioners.

A sophisticated signal priority control algorithm should be able to handle complex traffic and bus conditions as well as make decisions based on comprehensive judgment of the real

---

Manuscript submitted on February 28, 2005. This work was supported in part by the California PATH program under Grant 1402/B745-01.

Lefei Li is with the University of Arizona, Tucson, AZ 85721 USA (e-mail: lefeil@email.arizona.edu)

Wei-Hua Lin is with the University of Arizona, Tucson, AZ 85721 USA (phone: 1-520-621-6553; fax: 1-520-621-5555; e-mail: weilin@ sie.rizona.edu).

Hongchao Liu is with Texas Tech University, Lubbock, TX 79409 USA (e-mail: hongchao.liu@coe.ttu.edu).

time information of bus arrival time and queue status. In particular, priority requests from all intersection approaches need to be categorized in a manner that the right service sequence can be selected to obtain optimal benefits for buses while the impact to non-transit vehicles is minimized. Such an algorithm must address following issues.

First, an evaluation mechanism that judges the priority level of transit vehicles on conflicting approaches needs to be developed in the algorithm to replace the simple first-come-first-serve principle. Second, the status of the operation of transit vehicles should always be taken into consideration, only those buses behind schedule will receive priority service and to what extent depends on the lateness and occupancy of the bus as well as the traffic condition. Third, the algorithm should be able to make the right decision on priority type, priority degree (to what extent), and service sequence in a timely manner.

Provision of signal priority to handle multiple requests can be characterized with the following expression:

$$\Pr i = \alpha * q\_length + \frac{\beta}{t\_est + \delta} + \sum_{i=1}^{n} \gamma_i X_i$$

$$\alpha, \beta, \gamma_i \in [0,1]$$

where $\alpha, \beta, \gamma_i$ are weights addressing different concerns. *q_length* represents the queue length of the requesting transit vehicle's approach. The request with a longer queue length should be given a higher priority. *t_est* is the estimated time that the transit vehicle can clear the intersection. If the current signal is green, then *t_est* is the estimated time for the transit vehicle to clear the intersection at the current speed. If the current signal is red and the transit vehicle is stopped, then *t_est* is the estimated time for the transit vehicle to start and clear the intersection. The request that need less time to clear the intersection will gain more weight. This strategy is similar to the deadline driven scheduling strategy of a multi task embedded system. The accuracy of *q_length* and *t_est* is critical to the performance of the system, which can be achieved with a queue monitoring and transit vehicle tracking system. $\delta$ is introduced to avoid the situation in which *t_est* could be quite small. $X_i$ is the other factor that should be taken into consideration, such as the status of schedule adherence and the traffic condition of bus routes.

In the following we will show that the various control discussed above can be effectively realized with CPN.

## III. CPN FOR TRAFFIC SIGNAL PRIORITY CONTROL

### A. *Advantages of CPN over PN and HPN*

In CPN, color refers to the type of data associated with tokens. In other words, tokens can have arbitrary values determined by their type or color [5]. Each place in CPN has an associated color set, which constrains the number and color of tokens that may move along the arc. The major advantage of CPN over PN is the programming capability, such as the graphical tool NETMAN described in [10]. With the definition of data types (colors), states and transitions, as well as messages and other parameters can be delivered through the CPN model. With this, CPN is capable of modeling large complex communication networks that are difficult to model with PN. A comparison of CPN and PN is displayed in Table I.

Table I. Comparison of PN and CPN

| | PN | CPN | Detailed Advantage |
|---|---|---|---|
| Complex System | ● | ●● | Compact Model |
| Programming Language | ● | ●● | |
| Hierarchical Structure | ● | ●● | |
| Simulation Performance | ●● | ●● | Performance Analysis for Large Systems |
| Visual | ● | ●● | CPN use domain concepts |
| Time | ●● | ●● | |
| State Space | ●● | ●● | |

Note: the number of dots corresponds to the sophistication level.

HPN is composed of a "classical" (discrete) PN and a continuous PN, the "fluid" version of usual timed PNs [4]. The time concept can be conveniently incorporated into CPN to provide performance analysis capability. In terms of compatibility, CPN can work with PN or HPN without much modification. Furthermore, when a traffic signal priority control system is incorporated into a network that requires more frequent interactions among system components, the advantage of CPN will become more pronounced.

### B. *System architecture of CPN for signal priority control*

Fig.1 depicts the logical architecture of a signal controller which consists of three components: logic control, priority control, and optimal control. Communication between the three components is critical for synchronization and resource sharing. CPN is powerful in modeling concurrent systems and handling the transition logic of traffic signals.

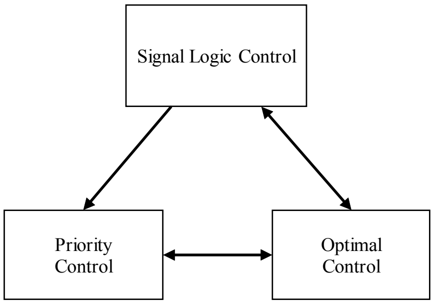

Fig. 1: System Structure of Traffic Signal Controller

The PN model of logic control of traffic signal has been developed in [2, 3]. In [6], a CPN model for traffic signal control was proposed. The optimal control requires real-time optimization. Many algorithms have been proposed for traffic signal planning, which will not be discussed here. In this paper, we will focus on 1) developing a CPN model for priority/preemption control and 2) developing interfaces with previously developed PN models of logic control of traffic signal and the optimal control component. The proposed CPN model is given in Fig. 2. In the following, we discuss in detail several major features associated with the proposed CPN model.

*1) Multiple requests*

Existing signal priority/preemption control algorithms cannot handle the situation in which buses call priority service from multiple intersections approaches. This incapability and the associated first-come-first service principle will result in extra delay to overall system. On the other hand, neither is a desirable strategy to provide priority at the same level to all buses. In our framework, an evaluation procedure is designed to weigh each request based on different hierarchy based on (1). Control factors, such as street types and queue lengths, are defined and embedded in the evaluation algorithm. Requests with higher priority will preempt requests with lower priority. This kind of preemption design in the CPN model is first proposed in [9] for scheduling problems.

*2) Real-time response to new requests*

One common approach to evaluate priority for multiple requests is to build some algorithms that evaluate all the requests. However, this may not be efficient since only the request with the highest priority is favored in signal plan generation. In our model, the new request will be compared only with the current highest priority request. If the new request is evaluated to have higher priority than the current one, the preemption will occur. Otherwise, the new request will be put on the waiting list until the current highest priority request is executed.

*3) Robust to changes in strategies*

The critical part of the priority control is the priority evaluation and signal plan generation. There are many strategies proposed to get the best result. In the proposed CPN model, the priority evaluation and signal plan generation are flexible, which can be easily interfaced with algorithms without any major changes to the overall structure.

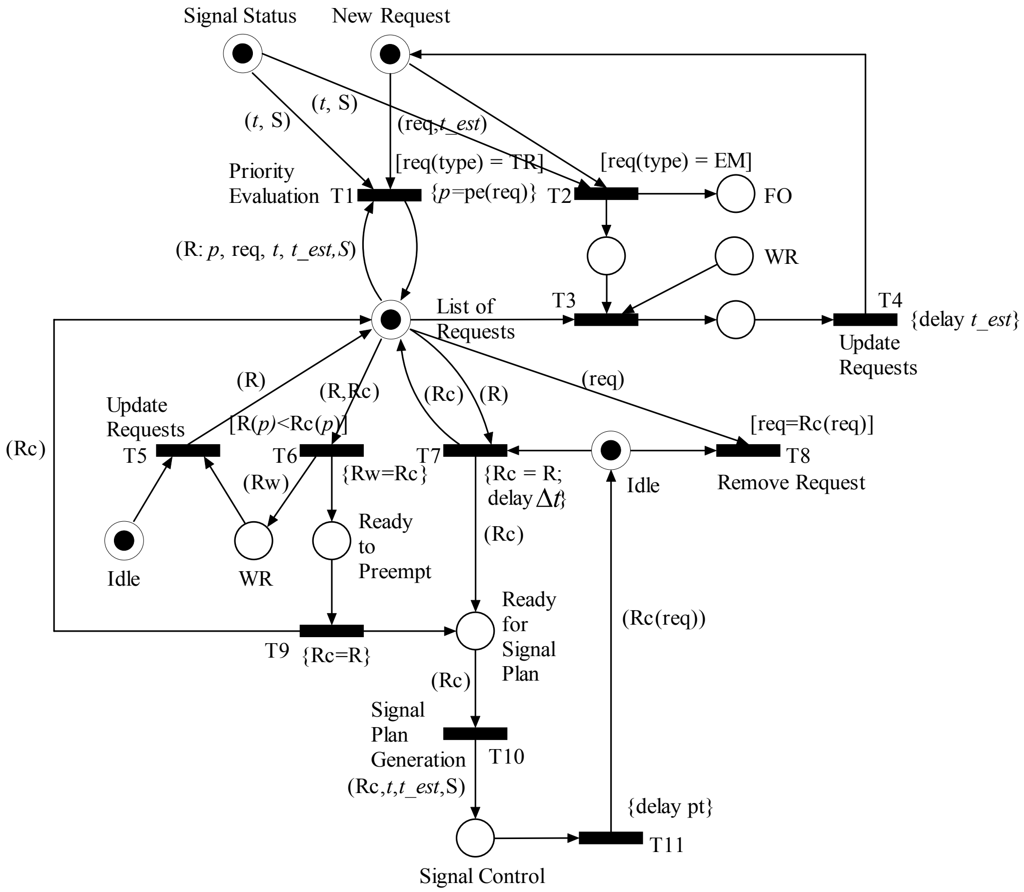

Fig. 2: A CPN Model of Traffic Signal Priority/Preemption Control

### C. Color Sets and Functions of the Proposed CPN

The basic Color Sets for the CPN model are:

Color t = int / time elapsed in current the signal phase

Color S = int [n] /signal sequence variables, such as signal id, signal phase, max green time, etc.

Color req = int [n] /request variables, such as request type (TR for transit vehicle, EM for emergent vehicle)

Color *t_est* = int /estimated time for vehicle passing the intersection

Color *p* = int /priority value

Color List = list req /requests list
Color R = union $p$ : req : t : $t\_est$ : S /combined request information (Rc is the current R)

$t$ is an integer variable indicate the passed time of the current signal phase. This is used to calculate the rest time of current phase. In [4], $\tau$ is used to represent the current time in the continuous part of the HPN model. In our proposed model, with the advantage of CPN, $t$ is employed with tokens to 'mark' the system, in other words, to represent the status of the system.

S is vector that contains all the signal information. It is a union of signal sequence variables such as signal id, signal phase, max green time, etc. This structure could be modified based on the requirement of specific control strategies. This kind of structure can be easily defined for CPN model. The modification of the signal sequence variables is performed within the vector S without severe effect to the structure of the whole CPN model.

$P$ is an integer priority value which is assigned to requests by 'priority evaluation' module, which will be discussed in the functional description of the CPN model.

$t\_est$ is a key parameter in signal priority control since the estimated passing time of a requesting transit vehicle is crucial to determine the priority signal plan.

R or $R_c$ is a union of a request (includes the priority value assigned to the request and estimate passing time of the request) and the system status information, such as $t$.

A CPN model of traffic signal priority/preemption control is shown in Fig. 2. It has six major components: emergent request, priority evaluation, preemption control, signal planning, update requests and list maintenance.

### *1) Request from Emergency Vehicles*

Upon receiving a priority request, the system will check the type of the vehicle making the request. The type is defined in the Color Set that can be easily modified in CPN. As shown in Fig. 2, if it is an emergency vehicle and the current signal of that direction is red, then T2 will be fired and a force off (FO) signal will be sent to the logic control system to change the crossroad green signal to red immediately. If the current signal is green, the FO signal will be used to keep the green phase until the emergent vehicle passes the intersection. Meanwhile, all the accepted requests of transit vehicles are terminated by T3. After the emergency vehicles clear the intersection, those requests will be updated and treated as new requests by the firing of T4.

Note that in PN the models of the requests from different types of vehicle have to be developed separately followed by invariant analyses to guarantee the consistency. The size of the model may grow rapidly with the increasing complexity of the system. CPN can represent a complex system in a more compact manner.

### *2) Priority Evaluation*

If the request is from transit vehicles, it will be evaluated with the current highest priority request by a priority evaluation function pe (req) of T1. If there is no accepted priority request, then it will be added to the list of requests with certain priority $p$. Since the firing of T1 requires tokens in all input places, an empty token should be placed in the 'List of Requests' place in the initial marking. Usually, the fire condition of a transition in CPN can be represented as a Boolean statement, which is very convenient for simple priority assignment mechanism. For complex priority assignment mechanism discussed in previous sections, the hierarchical structure of CPN model provides the capability to build a sub-page CPN for the 'priority evaluation' module, making it easy to make strategy modification. In order to change the control strategy of the system, one can simply modify the sub-page CPN of 'priority evaluation'.

### *3) Preemption Control*

T6 and T7 are two competing transitions that represent the preemption situation and the normal situation.

T7 will be fired if the signal planning system is idle. The highest priority request in the request list will get ready for a corresponding signal plan. The token in the 'Idle' place will be removed and a token Rc will be placed in the 'Ready for Signal Plan' place. Also, a token will be placed back to the list, meaning that the request is subject to a comparison with other existing requests.

T6 will be fired if a new request has a higher priority than the current processing request (Rc). It will lead to the preemption operation. Rc will be stored in Rw, which will go to the waiting list of requests (WR). Then, T9 will make the new request preempt. The processing request will become the new Rc for signal plan generation.

When the highest priority request is fulfilled, the token in the 'idle' place will fire T5, then the request stored in WR will get updated and go back to the list of requests. Idle place is the major design mechanism in PN and CPN to achieve synchronization and avoid conflict.

### *4) Signal Planning*

As long as a token is placed into 'Ready for Signal Plan' place, T10 will be fired to generate the signal plan in favor of the request. With the signal plan, an estimated time for the transit vehicle to pass the intersection (pt) is calculated. T11 will hold the system for pt seconds before other lower priority requests can be processed. This component is similar to 'priority evaluation'. A sub-page may be developed to achieve some sophisticated mechanism for generating signal plans. However, the easiest way of changing the signal plan in favor of transit vehicle is the modification of max green time of certain signal phases. This can be defined by a Boolean statement.

### *5) Update Requests*

Since emergency vehicles will always preempt the current signal plan, the request from an emergency vehicle will 'block' the transit vehicle's priority control. This requires a recovery procedure for the transit vehicle requests. In the proposed CPN model, an 'update requests' module is designed to handle this situation. After estimated the passing time of an emergency vehicle, the requests of transit vehicle will be updated according to the current status of the transit vehicle. In this case, the estimated passing time of the transit vehicles is recalculated. Then the requests are treated as new requests. With this component, the statuses of transit vehicles are updated to keep the benefit of the generated signal plan. It makes no sense to provide priority to a transit vehicle that just cleared the intersection.

### *6) List Maintenance*

After a request is processed, T8 will be fired to remove the request from the list. CPN provides the ability to define and operate list. PN can only have multiple tokens within a place, but the tokens are identical without type. Therefore, much more information is carried through a CPN model, making the representation of systems more flexible and powerful.

In a functionality view, three key components have to be defined: the priority evaluation function *pe (req)*, signal plan generation strategy, and the estimated finish time *pt*.

The priority evaluation function *pe (req)* assigns requests with different priorities according to the expression given in Sec. II. Signal plan generation has been proposed with many approaches. Such as the integer programming algorithm based on queue information in [4] and the total delay evaluation of candidate signal plans method in [8]. The estimated finish time *pt* is the time duration that the processed request vehicle can pass the intersection with the generated signal plan. It is a function of *g_max, g_min, t and t_est*.

## IV. AN EXAMPLE FOR QUEUE JUMP PHASES

A more aggressive method for bus priority at signalized intersections is the queue-jump lane that allows bus to move to the front of queue by using the right-turn lane or bus only lane. Queue-jumper is often associated with a traffic signal that turns green a few seconds ahead of the other signals with special indication of right-of-way for buses. It is a low-cost approach because the existing right turn lane can be used to provide such bypass opportunities for buses. In this particular case, we need to determine if a special phase exclusively for bus movements needs to be added. This should be considered in conjunction with green extension and red truncation. A phase list structure can be defined in the color set. Table II is a simple example of structure and value for the signal phases for a four way intersection without left-turn.

Table II. An example of the structure and value list for signal phases

| Phase* | mov | max_G (sec) | min_G (sec) | next |
|---|---|---|---|---|
| 0 | 1, 3 | 90 | 40 | 1 |
| 1 | 2, 4 | 60 | 20 | 0 |
| S* | Open* | Open | Open | Open |

* 'Phase': The traffic signal phase. 'S': the special phase.

** 'open' represents the situation in which the parameters will be determined based on control strategies on a real-time basis.

The following modifications should be made to the color set:

mov: The enabled movements of the signal phase.
max_G: The maximum green time of the signal phase.
min_G: The minimum green time of the signal phase.
next: The next signal phase.

Ph_C: Current signal phase.
Ph_temp: Temporary signal phase.

The queue-jump phase strategy is in a way similar to the special phase strategy that is often used in traffic signal control. The difference is that the special phase in queue-jump phase strategy is corresponding to a 'physical' special signal for transit vehicles. A possible modification is an extra condition check, which will examine whether there exists considerable queue length ahead of transit vehicles or not. If there are no vehicles before the transit vehicle, then there is no need to enable the special phase.

The sub-page CPN model for the special phase strategy is created (see Fig. 3) that decides which basic strategy, green extension ('green_ext') or red truncation ('red_trunc'), should be employed. Suppose the traffic light is currently red. The resulting transitions are as follows:

T1: The objective is to decide whether to perform red truncation or not. We need to estimate the waiting time before truncation can be performed. It is represented by $\Delta t$ . The constraint here is that the truncation cannot be performed until the minimum green time of the current phase is satisfied.

T2: Get the information of the next signal phase according to the normal sequence.

T3: When the next signal phase is not serving the transit vehicle's approaching movement, then insert a special phase to serve the transit vehicle. In the queue jump case, the special phase is dedicated to the special signal for transit vehicle. The time of the phase is $t_{est}$. Here $\tau$ is added to max_G to reflect the tolerance of errors associated with the estimation of the transit vehicle's passing time. Set the special phase's 'next' phase as the original 'next' phase in normal sequence. Then the special phase is set as the current phase.

T4: When the next signal phase is serving the transit vehicle's approaching movement, it is a normal red truncation. The only action is to set the next phase as the current phase to perform 'truncation'. Since the transit vehicle can 'jump' to the front of the queue, another option for the queue jump strategy is to make the next phase into special phase for transit vehicles. The transit vehicle will then have higher priority over other vehicles in the same approach. This option is not included in Fig. 3.

T8: We need to calculate delay $\Delta t$ and then perform the truncation or the special phase.

The right branch is for green extension, which is similar to the case for red truncation except that the calculation is for the extended green time. We will not discuss this branch in detail.

Note that other strategies, such as the phase suppression strategy, can be modeled in the same way. Skip a non-priority phase is similar to inserting a special phase that is equivalent to the current green priority phase. The only modification is the source of demand information for the non-priority phase. In our proposed system, this information is maintained in the 'Optimal Control' part.

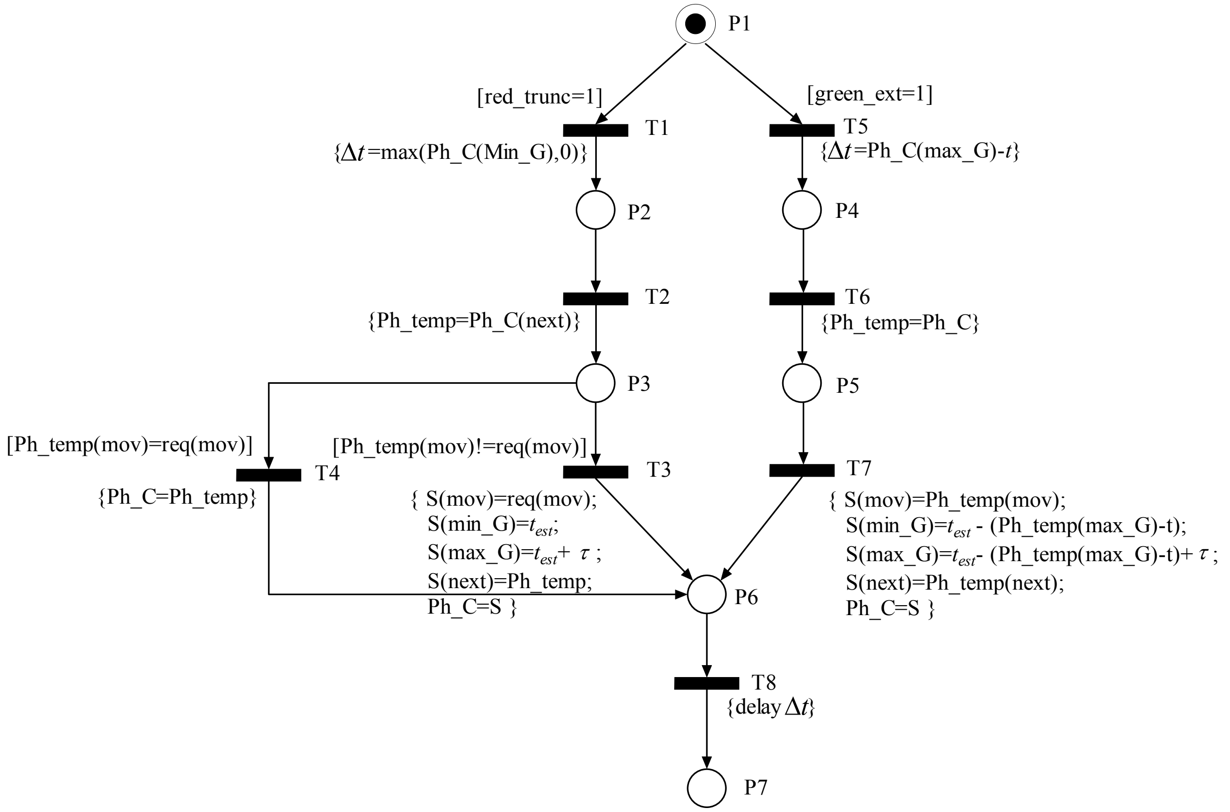

Fig. 3: The sub-page CPN model for the queue-jump phase

## V. CONCLUSION AND FUTURE WORK

The design of a Colored Petri Nets (CPN) model for traffic signal priority/preemption control is discussed to handle multiple requests. CPN is an extension of PN that has a compact and hierarchical representation that is particularly useful to traffic signal priority control for transit vehicles. The robustness of the CPN model to strategy changes makes it convenient for real-time applications.

When signal controllers are incorporated into a traffic network, numerous communication tasks will make the system extremely complex. Building the system without powerful validation of the design may lead to unexpected cost even failure. This leads to the requirement of model analysis tools. One of advantages of using CPN for system modeling is that CPN encompasses powerful analysis tools.

Full occurrence graph (O-graph) is one of the analysis methods of CPN. It can be employed to verify all the dynamic properties (i.e., reachability, boundedness, home, liveness, and fairness) associated with a CPN model. The construction of O-graphs and the associated verification of dynamic properties can be fully automated. This means that O-graphs provide a very straightforward and easy-to-use method to analyze the properties of a given CPN [5]. Invariants analysis is widely used in PN analysis. CPN also has this analysis tool. Work is ongoing to explore the O-graph analysis and invariant analysis for validating the traffic signal priority/preemption control CPN model.

## REFERENCES

[1] F. DiCesare, P. Kulp, M. Gile, and G. F. List, “The Application of Petri Nets to the Modeling, Analysis and Control of Intelligent Urban Traffic Networks,” in Proc. 15th Int. Conf. Application Theory Petri Nets, R.Valette, Ed., Zaragoza, Spain, June 1994, pp. 2–15.

[2] G. F. List and M. Cetin, “Modeling Traffic Signal Control Using Petri Nets”, *IEEE Trans.Intelligent Transportation Systems*, vol. 5, no. 3, pp. 177–187, Sep. 2004.

[3] J. Gallego, J. Farges and J. Henry, “Design by Petri Nets of an Intersection Signal Controller”, *Transpn. Res.C*, vol. 4, no. 4, pp. 231–248, 1996

[4] A. Di Febbraro, D. Giglio and N. Sacco, “Urban Traffic Control Structure Based on Hybrid Petri Nets”, *IEEE Trans. Intelligent Transportation Systems*, vol. 5, no. 4, pp. 224–237, Dec. 2004.

[5] K. Jensen, “*Coloured Petri Nets --- Basic Concepts, Analysis Methods and Practical Use*”, vol. 2, Springer-Verlag New York Berlin Heidelberg, 1995

[6] A. Perkusich, L.M. de Araujo, R. de S. Coelho, K.C. Gorgon and A.J.P. Lemos, “Design and Animation of Colored Petri Nets Models for Traffic Signals”, CPN Workshop, Aarhus Denmark, Oct. 1999.

[7] A. Skabardonis, “Control Strategies for Transit Priority”, *Intellimotion*, vol. 7, no. 2, pp. 6-11, 1998

[8] S. Yagar and B. Han, “Procedure for Real-time Signal Control that Considers Transit Interference and Priority”, *Transpn. Res. B*, vol. 28B, no. 4, pp. 315-331, Aug. 1994

[9] M. Naedele, “Petri Net Models for Single Processor Real-Time Scheduling”, TIK-report, no. 6, Nov. 1998.

[10] F. DiCesare, M. Gile, and P. Kulp, “A Code Generation Tool for Simulation and Control, Based on Colored Petri Nets,” in Proc. IEEE Conf. Syst., Man, Cybern Intelligent Systems 21$^{st}$ Centrury, vol. 4, 1995, pp. 3063-3068.