LICP: A Look-ahead Intersection Control Policy with Intelligent Vehicles

Minjie Zhu, Xu Li, Hongyu Huang, Linghe Kong, Minglu Li, Min-You Wu
Department of Computer Science and Engineering
Shanghai Jiao Tong University

Abstract

We consider a practical application of intelligent vehicles for intersection traffic control. Specially, we study the intersection traffic control problem using Reservation-based Intersection Traffic Control System, which utilizes the information exchange between intelligent vehicles and management agents around the intersections to direct traffic, instead of traffic lights. We focus on how to design an effective passing permission (PP) allocation strategy for this system. In this work, with an observation that will cause this system to be inefficient, we propose a novel look-ahead passing permission allocation strategy (LICP) for intersection traffic control. The large-scale testing results show that LICP can make nearly 25% performance improvement on average intersection delay than the previous First Come, First Serve method (FCFS).

Keywords *- Intersection Traffic Control, Intelligent Vehicle System, Passing Permission, Look-ahead Intersection Control Policy*

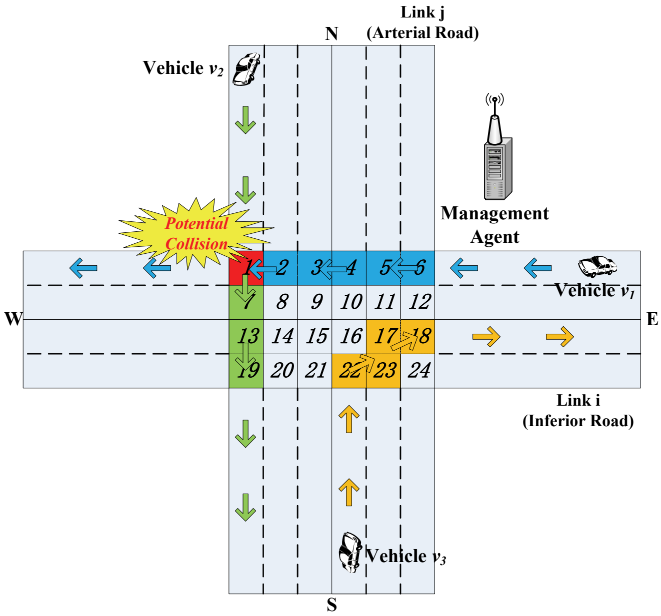

Figure 1. An illustration of intersection model

1. Introduction

Traffic congestion is widely distributed around the road networks in urban areas. According to a study of 85 Unite State cities [8], every driver spends 46 hours/year on average waiting in traffic jams that leads to a low efficient transportation system, considerable gasoline waste and air pollution. Over last thirty years, urban traffic management has attracted tremendous interests from both government and academic[13][14][15]. Especially, for the intersection traffic control topic, a large number of works have been carried out which aim to minimize the average intersection delay for the vehicles, such as optimal traffic light scheduling [5][6][7].

Recently, vehicles can be easily equipped with powerful processing units and wireless transmitters to assist route planning and autonomous steering [9][10][11], such as GPS navigation system, various sensors, DSRC interface, etc. By utilizing the information exchange between intelligent vehicles and management agents around the intersections, several previous works [12] focus on how to design effective intersection traffic control policies with no traffic lights involved. Specially, K. Dresner introduced Reservation-based Intersection Traffic Control System [1][2][3][4]. In this system, an intersection area is divided into grids. A vehicle needs to apply a *passing permission* (*PP*) from the *management agent* (MA) of its next intersection to avoid collision before it passes this intersection [1]. To be more precisely, a passing permission is correlated to a set of grid unoccupied by other passing permissions in spatial-temporal space. In the meantime, these grids will constitute a path for the vehicle to go through the intersection area, as shown in Figure 1. In K. Dresner’s work, the author proposed that how to allocate the passing permissions is based on the First Come, First Serve methodology (*FCFS*) [2][4].

In our work, with an observation about the inefficiency of FCFS, we introduce a new policy for this reservation-based intersection traffic control system. Concretely, we design a look-ahead passing permission allocation strategy to overcome the disadvantages of FCFS method mentioned above. The large-scale testing results show that our method can make 25% performance improvement on average intersection delay than the previous FCFS method. Overall, this paper is not concerned with details of the networking aspect, but primarily with a practical application of intelligent vehicles for intersection traffic control.

The remainder of the paper is organized as follows. Section 2 introduces the intersection model and problem statement. In Section 3, we review the reservation-based intersection traffic control system and the FCFS policy. We

978-1-4244-5113-5/09/$25.00 ©2009 IEEE

propose a novel look-ahead control policy in Section 4. Section 5 is the performance evaluation and testing results. Section 6 concludes the paper.

2. Model and Problem Statement

2.1. Model

In reservation-based intersection traffic control system, we consider a model that an intersection is connected with 4 bidirectional links with same length but different number of lanes (to represent the arterial and inferior links). As shown in Figure 1, arterial link has six lanes while inferior link only has four lanes. Here, an implied fact is that links can have different traffic capacities and we will consider the uneven traffic flow on different links based on this model. An intersection area is divided into grids (In Figure 1, there are 6×4 grids in the intersection area).

We assume that all the vehicles are driven by micro computers and there is an arbiter agent named *management agent* (MA) at each intersection, which is responsible for the traffic control, i.e., for the *PP* allocation to avoid the potential collision shown in Figure 1.

When an intelligent vehicle enters into a link (e.g., vehicle v_1 enters into link i at *E* point in Figure 1), it needs to send its real-time status information to the MA of its next intersection to apply a passing permission, only with which the vehicle can go through the intersection area. At the same time, the vehicle cannot stop in the intersection area and must follow the path, which is constituted by a set of grids occupied by this passing permission (yellow grids for vehicle v_3 in Figure 1). In this model, there are no traffic lights involved. The real-time status information of a vehicle includes vehicle id (*vid*), geographical coordinates (*pos*), current speed (*v*) and travel direction (*td*) and the time when it will arrive at its next intersection.

2.2. Problem Statement and Performance Metrics

Based on this model, our objective is to design an optimal passing permission allocation strategy to minimize the intersection delay with the real-time status information of intelligent vehicles.

We use intersection delay as a metric to evaluate the performances of different intersection traffic control policies. We first give the definition of intersection delay for one vehicle: For vehicle v_i, we define its intersection delay as $t(i)-t_0(i)$, where $t(i)$ is the total time cost between two positions including the intersection delay (E.g., A path from position S to E for vehicle v_3 in Figure 1). $t_0(i)$ is the optimal time cost if the vehicle can travel between two positions with no intersection delay.

Globally, the *average delay* of an intersection [1] is defined as:

$$\frac{1}{|C|}\sum_{v_i \in C}(t(i)-t_0(i)) \qquad (1)$$

FCFS Policy:

1) Initialization: Let *granularity* ←(n_1, n_2)($n_1, n_2 \in N^+$), and v_{max} ← r ($r \in R^+$);
// v_{max} is the max speed of the intersection
2) Processing requests:
flag ← true;
// $L(r_i)$ is the list of requests received by policy
While $L(r_i)$ is not empty
  r ← the first element in $L(r_i)$;
  path ← calculate grids needed by *r* using (*granularity*, *pos*, *td*);
  for *i* ← 1 to num(*path*)
    cv ← choose a proper value from *v* and v_{max};
    time[*i*] ←the expected time for *path*[*i*] by *cv*;
    if *time*[*i*] conflicts with accepted request
      flag ← false;
      break;
    end if
  end for
  if *flag* = true
    generate a passing permission for *r*;
  else
    generate a rejection for *r*;
  end if
end while

Figure 2. The Algorithm of FCFS

where *C* is the set of all vehicles passing the intersection during a given period of time (Here we set it as the simulation time).

Another metric is *maximum delay* of an intersection [1], which is defined as:

$$\max_{v_i \in C}(t(i)-t_0(i)) \qquad (2)$$

3. The Overview of FCFS Policy

In this section, we review the FCFS policy proposed by Dresner [1][2][3][4]. The main idea of FCFS is that a vehicle will get a passing permission if and only if the MA of its next intersection can find a *feasible grid-based path*, which is not conflicted with other paths/grids (i.e., not occupied by other passing permissions) in spatial-temporal space (for collision avoidance), by utilizing the real-time status information of vehicles. Otherwise, the request for a passing permission will be rejected by MA. The detailed FCFS policy is presented as a procedure in Figure 2.

4. Look-ahead Intersection Control Policy (LICP)

4.1. Observation

Here, we give a simple example to show the inefficiency of FCFS policy.

In Figure 1, vehicle v_1 is running on link i from *E* to *W* and vehicle v_2 is running on link j from *N* to *S*. Figure 3(a) shows their individual reservations and potential conflict. To be more precisely, vehicle v_1 wants to use Grid 1 during [t_3,

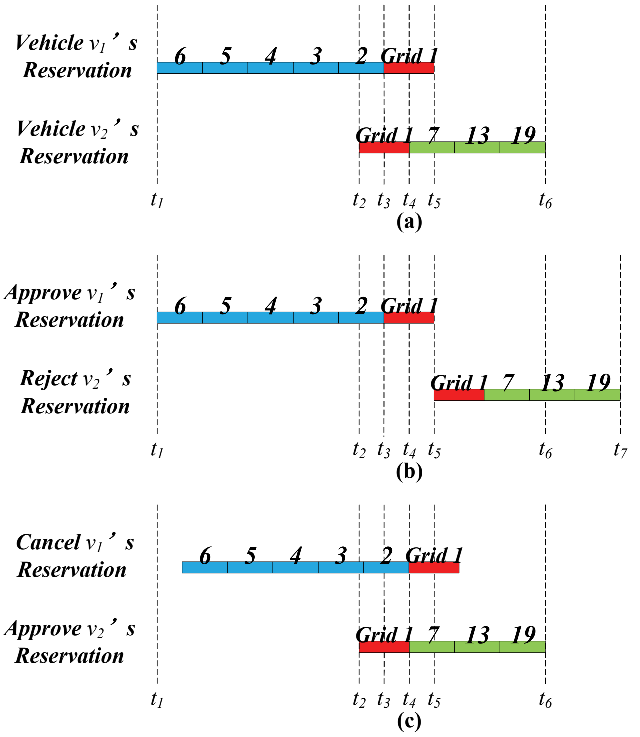

Figure 3. Reservations under different situations

t_5], and vehicle v_2 wants to use Grid 1 during [t_2, t_4]. Thus, there is a time overlap during [t_3, t_4], which leads to a conflict. Here, we assume that vehicle v_1 sends its passing permission request earlier than vehicle v_2.

Based on FCFS policy, MA will accept the passing permission request of vehicle v_1 and postpone vehicle v_2's request until t_5, as shown in Figure 3(b). In this situation, the total intersection delay of v_1 and v_2 is t_5- t_2.

Figure 3(c) shows another solution for this scenario not based on FCFS policy. That is, MA cancels/postpones the reservation of v_1 and approves path reservation of v_2 even if v_2 sends its request later than v_1. In this situation, the total intersection delay of v_1 and v_2 is t_4- t_3, here t_5-t_2>t_4-t_3. Based on this observation, we tend to design a more efficient *PP* allocation strategy than the current FCFS policy.

4.2. Look-ahead Intersection Control Policy (LICP)

In this section, we introduce the Look-ahead Intersection Control Policy (LICP) to improve the performance of the Reservation-based Intersection Traffic Control System.

The main idea of LICP is choosing a right decision of *PP* allocation to reduce the average intersection delay based on two criteria.

The first criteria is $D_{postpone}(r)$, which is the predictive value of total delay if postponing the current reservation request r, as shown in Equ.(3):

$$D_{postpone}(r)=(m+1)\times D(r)+t \qquad (3)$$

where $D(r)$ is the delay of vehicle v_i due to postponing its request r to avoid conflict with other earlier requests. m is the number of vehicles following v_i, which is to represent the consecutive effect of delay on the vehicle queue. Here, t is another parameter, which is used to solve the starvation problem, as shown in Equ.(4):

$$t=\begin{cases}0 & \text{if time counter } tc \text{ is not activated}\\ s & \text{if time counter } tc \text{ is activated}\end{cases} \qquad (4)$$

where s is the current value recorded by the time counter. Here, MA keep a time counter tc for each vehicle (More discussion about Eq.(4) will be presented later).

The second criteria is $D_{allow}(r)$, which is the predictive value of total delay if approving the current reservation r, as shown in Equ.(5):

$$\begin{cases}L_i=Sort(L_c, f_i) & f_i\in F \text{ and } i\in[1, |F|]\\ D_{allow,\,i}=\sum_{j=1}^{n} D_{postpone}(r_j) & n=|L_i| \text{ and } r_j\in L_i\\ D_{allow}(r)=\min\{D_{allow,\,i}\mid i\in[1,|F|]\}\end{cases} \qquad (5)$$

where L_c is the list of approved reservations conflicted with r. To be more precisely, these approved reservations need to be canceled and postponed temporarily because we consider the current reservation request r has been approved. F is a set of sorting functions for L_c and f_i is a function in F.

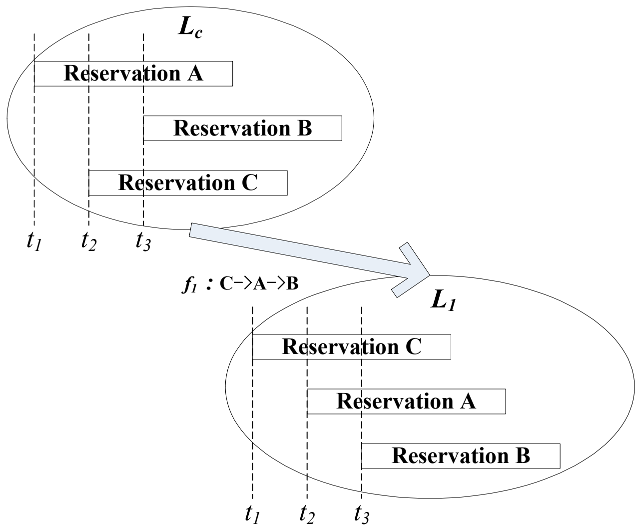

Figure 4. An Example of L_c

For instance, as shown in Figure 4, L_c has three approved reservations A, B and C. We define the output of f_1 (denoted as L_1) on L_c is the postponing order: C→A→B. Accordingly, based on this order, we define that $D_{allow,\,i}$ is the total delay of L_i. Due to the fact that various postponing orders lead to the different total delays, LICP will choose f_i with minimum $D_{allow,\,i}$, denoted as $D_{allow}(r)$.

Overall, in LICP scheme, when a vehicle sends a *PP* request r, MA will first check whether it conflicts with the approved reservations. If no conflict exists, MA will approve the request r and allocate a new passing permission. Otherwise, it will calculate the values of $D_{postpone}(r)$ and

Look-ahead Intersection Control Policy:

1) Initialization: Let granularity ← (n_1, n_2)(n_1, $n_2 \in N^+$), v_{max} ← r ($r \in R^+$) and *rules* ←Φ(*sort rules*);

2) Processing requests:

flag ← true;
// $L(r_i)$ is the list of requests received by policy
While $L(r_i)$ is not empty
  r ← the first element in $L(r_i)$;
  path← calculate grids needed by r using
    (*granularity* , *pos*, *td*);
  for i ← 1 to num(*path*)
    cv ← choose a proper value from v and v_{max};
    time[i] ← the expected time for *path*[i] by cv;
    if *time*[i] conflicts with accepted requests
      add all the requests conflict with r to list L_c;
    end if
  end for
  if L_c is not empty
    if vehicle has received 10 replies of postponement
      start *timer*;
    end if
    if time counter tc is activated
      t ← s;
    else
      t ← 0;
    end if
    $D_{postpone}(r)$ ← calculate the delay of postponing
        request r using parameter t;
    for i ← 1 to num(*rules*)
      sort L_c using *rules*[i];
      $d[i]$ ← calculate the total delay of all the requests
        in L_c;
    end for
    $D_{allow}(r)$ ← min($d[i]$);
    if $D_{postpone}(r) > D_{allow}(r)$
      modify each request in L_c and notify the vehicles;
    else
      flag ← false;
    end if
  end if
  if *flag* = true
    generate a passing permission for r;
    close time counter tc;
  else
    generate a postponing reply for r;
  end if
end while

Figure 5. The Algorithm of LICP

$D_{allow}(r)$. If $D_{postpone}(r) > D_{allow}(r)$, which means postponing the current request r will have more total delay than approving it, then MA will approve it. If $D_{postpone}(r) \leqslant D_{allow}(r)$, which means approving the current request r will have more total delay than postponing it, then MA will postpone it.

In addition, we give more discussion about Equ.(4), which is designed for solving starvation problem. In a given intersection, for vehicle v_i, if MA consecutively postponed its reservation request for ten times, the time counter tc_i will be activated. Accordingly, the value of s will increase with time, which contributes to calculate $D_{postpone}(r)$. Finally, v_i will get a passing permission because of

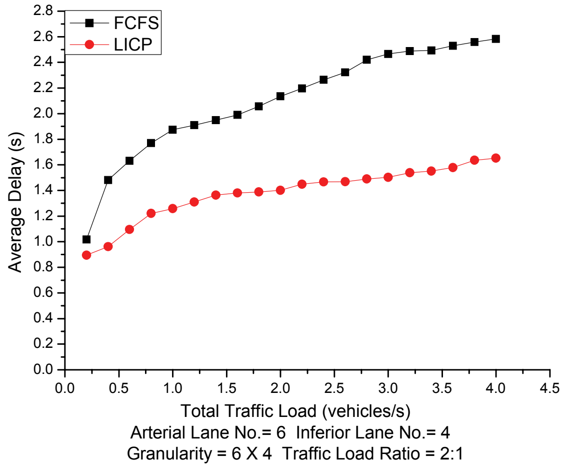

Figure 6. The average delay of vehicles while using different control policies

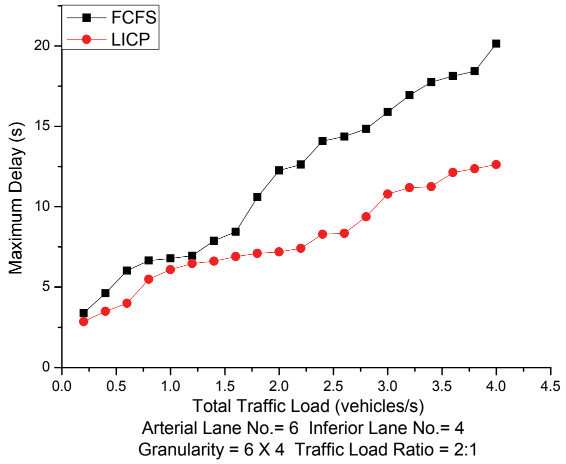

Figure 7. The maximum delay of vehicles while using different control policies

$D_{postpone}(r) > D_{allow}(r)$. At the meantime, tc_i will be stopped and the value of s will be reset with 0.

The detailed LICP algorithm is shown in Figure 5.

5. Performance Evaluation

5.1. Simulator

We develop a simulator to evaluate the performances of FCFS and LICP. It can model the individual behavior of each vehicle and the realistic scenario in city urban area, such as arterial and inferior rods, uneven traffic loads on different links, etc, as shown in Figure 1. In our simulation, we set the arterial link with six lanes while inferior link with four lanes. Total Traffic Load (including all the traffic on the links) means how many vehicles enter into the system per second during simulation, which can be used to model the rush/unrush hours. At the meanwhile, we introduce another parameter Traffic Load Ratio, which is the ratio of traffic loads on arterial and inferior links in order to represent

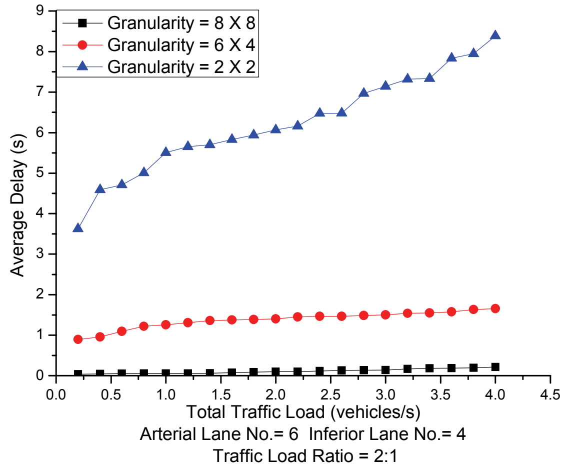

Figure 8. The average delay of vehicles while using different granularities

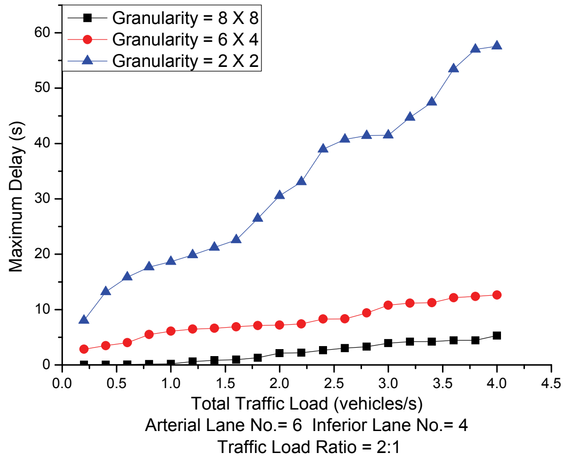

Figure 9. The maximum delay of vehicles while using different granularities

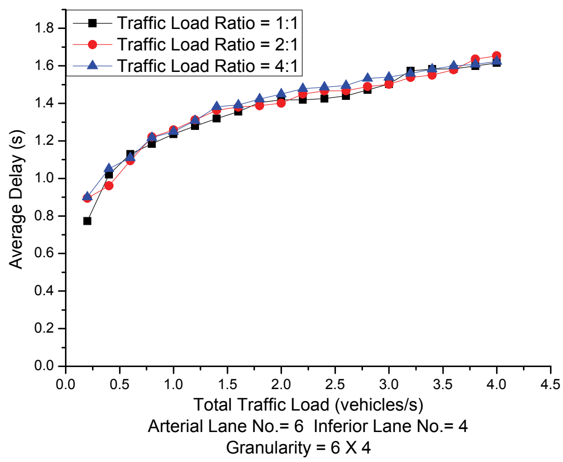

Figure 10. The average delay of vehicles while the traffic load assignment is different

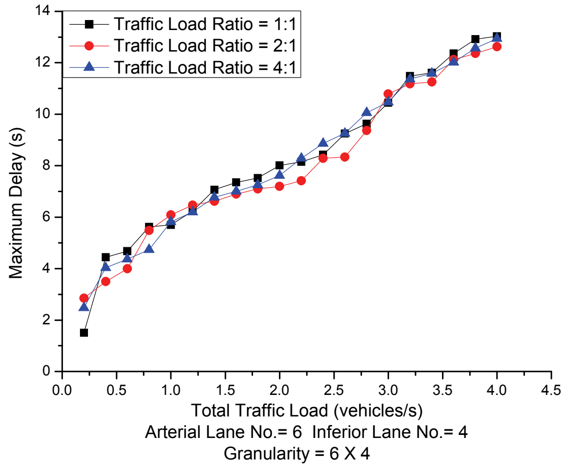

Figure 11. The maximum delay of vehicles while the traffic load assignment is different

uneven traffic. In addition, we divide intersection area into $n_1 \times n_2$ grids, where n_1 and n_2 are configurable granularity parameters. (For instance, in Figure 1, the intersection area is divided into 6×4 grids, Total Traffic Load = 3 and Traffic Load Ratio = 2:1 means during simulation there are three vehicles enter the system in each second, especially two vehicles will be on arterial link (*link j*) and only one on inferior link (*link i*).

5.2. Numerical Results

5.2.1. LICP vs. FCFS. From Figure 6 and 7, we can see that LIPC outperforms FCFS in terms of two metrics, respectively. Generally, LICP can make nearly 25% performance improvement (delay reduction) over FCFS. We can also find that the performance improvement of LICP becomes larger with increasing of total traffic load. It can be explained that with light traffic load, the conflicts between passing permission requests seldom happened. With heavy traffic load, however, high density of passing permission request leads to frequent conflicts. In this situation, LICP will play a more important role for conflict avoidance and delay reduction.

5.2.2. The delay of vehicles. Figure 8 and 9 show the average delay and maximum delay of LICP under different grid granularities respectively.

Here, we tend to examine the effect of grid granularity on the performance of LICP. i.e. granularity = 6×4 means the intersection is divided into 24 grids (default value, as shown in Figure 1), and granularity = 8×8 means divided into 64 grids). From Figure 8 and 9, it can be seen that compared with three granularities, 8×8 setting always has the best performance and 2×2 setting has the worst performance. The potential reason is that with 2×2 setting, the intersection area is divided into only 4 grids, which leads to inefficient utilization of space. A passing permission will occupy at least two big grids for a vehicle. In such a situation, PP request will be always canceled/postponed, which will increase the average delay or maximum delay inevitably. In addition, we see that both average delay and maximum delay of the intersection become larger with increasing of total traffic load. This is because heavy traffic load means more vehicles intend to compete for the grids, so the delays will increase due to PP request conflict.

Next, we explore the performances of LICP with different traffic load ratios. In Figure 10, we can see that with the same granularity, LICP has similar performance with different traffic load ratios in terms of average delay and maximum delay, respectively. This is because in LICP scheme, we take the fairness into consideration to handle the uneven traffic and starvation avoidance. To be more precisely, with parameter t in Euq.(3), LICP will give a vehicle higher priority to obtain a passing permission if it already waited for a long time. In this situation, vehicles on the inferior links will not experience much longer delays than vehicles on the arterial links. At the same time, from Figure 11 we can see that no vehicle experienced unaccepted long delay, which is equal to starvation. It can be explained that with parameter t, which prevents the vehicles from being postponed consecutively, no starvation would happen in LICP scheme.

6. Conclusion

In this paper, we studied an intersection traffic control problem with intelligent vehicles. By utilizing the information exchange between the vehicles and management agents of intersections, we focus on how to allocate passing permissions among the vehicles to minimize the average intersection delay with no traffic light involved. With an important observation, we proposed look-ahead passing permission allocation strategy (LICP). From the large scale simulation, we demonstrate that LICP can make considerable performance improvement in terms of average intersection delay, compared with the previous First Come, First Serve (FCFS) strategy.

Several issues remain to be addressed further. For example, this work did not touch the details of the networking aspect, but primarily focus on a research issue in traffic domain. We still need to study the various topics in network domain, such as the communication cost of information exchange, extending the network model to enable the communication between the vehicles, not only between vehicles and management agents, etc. These works are currently in progress in our lab.

Acknowledgements

We thank anonymous reviewer for their helpful comments. Especially, we appreciate the precious comments and valuable suggestion from Dr. Gongjun Yan at Old Dominion University.

This research was partially supported by Natural Science Foundation of China grant No.60573138 and 60773091, the National Grand Fundamental Research 973 Program of China under Grant No.2006CB303000, the National 863 Program grant No.2005aa122320, and National Science Foundation of USA grant CNR-0626380.

References

[1] K. Dresner, and P. Stone. Multiagent traffic management: A reservation-based intersection control mechanism. *The Third International Joint Conference on Autonomous Agents and Multiagent Systems*, New York, USA, July 2004, pages 530–537.

[2] K. Dresner, and P. Stone. Multiagent Traffic Management: An Improved Intersection Control Mechanism. *The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems* (AAMAS 05), Utrecht, The Netherlands, July 2005, pages 471-477

[3] K. Dresner, and P. Stone. Multiagent traffic management: Opportunities for multiagent learning. K. Tuyls et al. (Ed.), LAMAS 2005, *Lecture Notes In Artificial Intelligence*, Springer Verlag, Berlin, 2006, pages 129-138.

[4] K. Dresner, and P. Stone. A Multiagent Approach to Autonomous Intersection Management. *Journal of Artificial Intelligence Research*, 2008, pages 591-656.

[5] M. Nakatsuyama, et al. Fuzzy Logic Phase Controller for Traffic Junctions in the One-way Arterial Road. *Proceedings of the IFAC Ninth Triennial World Congress*, Pergamon Press, Oxford, 1984, pages 2865-2870

[6] Liu Zhiyong, et al. Artificial neural networks self-tuning predictive control for traffic signals. *Control Theory & Applications*, 2003, 20(6), pages 933-937

[7] Fang F.C., Elefteriadou, L. Capability-Enhanced Microscopic Simulation With Real-Time Traffic Signal Control, Intelligent *Transportation Systems, IEEE Transactions,* Tokyo, Japan, Dec. 2008, pages 625 – 632.

[8] Texas Transportation Institute, 2004 urban mobility report, September 2004, accessed at http://mobility.tamu.edu/ums in December 2004.

[9] S. Rogers, C.-N. Flechter, and P. Langley. An adaptive interactive agent for route advice, *Proceedings of the Third International Conference on Autonomous Agents (Agents'99)*, Seattle, WA, USA, 1999, ACM Press, pages198–205.

[10] T. Schonberg, M. Ojala, J. Suomela, A. Torpo, and A. Halme. Positioning an autonomous off-road vehicle by using fused DGPS and inertial navigation. *2nd IFAC Conference on Intelligent Autonomous Vehicles*, 1995, pages 226–231.

[11] D. A. Pormerleau. *Neural Network Perception for Mobile Robot Guidance*. Kluwer Academic Publishers, 1993.

[12] Li Z., Sun D., Zhang Z., Song J., Xiao D.,Control Mechanism Analysis of Small-Agent Networks Using a Distinguished Node Model for Urban Traffic Controls. *Automation Science and Engineering, IEEE Transactions*. July 2008, pages 420-430.

[13] Xu Li, Wei Shu, Minglu Li, et al. Performance Evaluation of Vehicle-based Mobile Sensor Networks for Traffic Monitoring. *IEEE Transactions on Vehicular Technology*, Volume 58, Issue 4, May 2009 Page(s):1647 – 1653.

[14] Xu Li, Ming-Lu Li. A Practical Map-matching Algorithm for GPS-based Vehicular Networks in Shanghai Urban Area. *The IET Communications Conference on Wireless, Mobile and Sensor Networks*, Shanghai, China, 2007.

[15] Xu Li, Wei Shu, Minglu Li. Traffic Data Processing in Vehicular Sensor Networks, *the 17th International Conference on Computer Communications and Networks*, Virgin Islands, USA, August, 2008.