Proceedings of the 1993 Winter Simulation Conference
G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles (eds.)

SIMULATING NETWORK TRAFFIC FLOWS WITH A MASSIVELY PARALLEL COMPUTING ARCHITECTURE

Gang-Len Chang
Thanavat Junchaya

Department of Civil Engineering
University of Maryland
College Park, Maryland 20742, U.S.A.

ABSTRACT

This paper presents a traffic network simulation model for real-time applications in IVHS. The proposed model has incorporated three key features essential for IVHS operations: (1) the capability of simulating both freeways and surface street networks as an integrated network; (2) a path-processing capability for representing drivers' route choice behavior at an individual/vehicle level; and (3) the capability of simulating different subnetworks at different levels of detail so as to increase the execution speed for real-time operations. Three simulation methodologies are implemented in the real-time traffic simulation model to satisfy the above requirements. These three methodologies are macroparticle traffic simulation model (MPSM), modified MPSM (M-MPSM), and microscopic (MICRO) model. Due to the real-time operating requirement, the proposed simulation model has been parallelized to take advantage of the parallel computers since they can offer the required computational power at an economical cost/performance ratio. Several simulation experiments have been carried out to compare the execution speed of each methodology. The preliminary research results indicate that parallel computing architecture offers a very promising alternative for the implementation of real-time traffic simulation.

1. INTRODUCTION

The paper describes network traffic simulation methodologies and their implementions on a massively parallel computer so as to improve their execution time for real-time applications. The paper is organized as follows: in this section the proposed simulation methodologies are described in detail, the second section presents the parallel simulation models. Numerical results and computational performance are presented and discussed in Section 3, follow by the conclusions in the last section.

MPSM Model for Freeways

The MPSM (MacroParticle Simultion Model) was developed as a special-purpose, macroscopic highway corridor model to capture the complex dynamics of the traffic system, particularly the fluctuation of travel time associated with departure time and the time-dependent congestion pattern (Chang, Mahmassani, and Herman, 1985). It is able to track the speed and position of individual vehicles on a highway segment, without the computational complexities associated with a microscopic model.

The MPSM is a fixed-time traffic simulation model. Instead of modeling traffic as compressible fluid flow as used in MACK or FREFLO (Payne, 1978, 1979), it views traffic as vehicle groups or bunches, termed macroparticles. The physical position of those macroparticles are updated in accordance with the local speed determined by a speed-concentration relation. Thus, the concentration of each section can be updated at every time step by tracing the physical positions of the particles. The MPSM employs conservation and speed-concentration equations to govern the traffic flow evaluation. The time-varying concentration in each segment can be expressed as follows:

$$k_i^{t+1} = k_i^t + \left(\frac{1}{l_i \, \Delta X_i}\right)\left(M_i^{e,t+1} - M_i^{o,t+1} + N_i^{t+1}\right) \quad (1)$$

where
k_i^t = concentration in Section i during the t-th time step (vehicles per lane-mile);
l_i = number of lanes in Section i;
ΔX_i = length of Section i;

$M_i^{e,t+1}$ = vehicles entering Section i from the preceding section during a given time step Δt;
$M_i^{o,t+1}$ = vehicles moving onto the next section;
N_i^{t+1} = the number of vehicles generated minus those exiting from the same section during the t-th time step.

In the MPSM, the concentration in each section is updated with Equation (1) at the beginning of every time step, and it is assumed to remain constant over the interval [t, t+Δt]. The corresponding mean speed prevailing during this interval can then be obtained from the given speed-concentration relation. The function form used in MPSM is

$$v_i^t = (v_f - v_o)\,(1 - \frac{k_i^t}{k_o})^{\beta_i} + v_o \qquad (2)$$

where
v_i^t = the mean speed in Section i during the t-th time step;
v_f, v_o= the mean free-flow speed and minimum speed on the facility, respectively;
k_o = the maximum or jam concentration; and
β_i = a location dependent parameter.

Note that the above equation can also be modified to include additional terms to improve the accuracy of the model (Papageorgiou, 1983).

The physical position of each macroparticle is updated at the end of every simulation interval and stored as one of the attributes of that macroparticle. Using the speed-concentration relation, the new position of particle m at the end of interval t+1 will be advanced by a distance $d(m,t+1) = \Delta t \bullet v_i^{t+1}$. If the particle reaches the end of the section, it will move to the next section after checking the available capacity of the next section and the intersection signal control.

Since MPSM actually move vehicles either individually or in bunches according to the prevailing local speed, it offers a greater flexibility in modeling the effects of in-vehicle information systems and route guidance strategies compared to traditional macroscopic flow models.

A Modified MPSM Model for Medium Congested Urban Streets (M-MPSM)

While the above MPSM methodology models freeway flows quite satisfactory, it may not be able to model freeway sections or urban streets where the speeds across lanes are not quite uniform. Thus, Equation (1) has been modified to compute speed in each lane as follows:

$$k_{ij}^{t+1} = k_{ij}^t + (\frac{1}{\Delta X_i})(M_{ij}^{e,t+1} - M_{ij}^{o,t+1} + N_{ij}^{t+1} + L_{ij}^{e,t+1} - L_{ij}^{o,t+1})$$

where
j = lane index j in Section i;
$L_{ij}^{e,t+1}$ = vehicles entering Lane j from adjacent lanes during a given time step Δt;
$L_{ij}^{o,t+1}$ = vehicles leaving Lane j into adjacent lanes.

Accordingly, Equation (2) is also modified to reflect the resulting lane speed:

$$v_{ij}^t = (v_{fij} - v_{oij})\,(1 - \frac{k_{ij}^t}{k_{oij}})^{\beta_{ij}} + v_{oij}$$

Lane-Changing

The ability to differentiate speeds across lanes is more realistic than restricting all vehicles in the same link to follow the same speed. In addition, we will be able to incorporate lane-changing mechanism into the M-MPSM models. For instance, if a link has a large fraction of turning flows, vehicles that are in the turning lanes but wish to go straight may actually change to the neighboring lane so as to increase their speeds.

The proposed lane-change mechanism in the M-MPSM is macroscopic in nature. It does not check the actual location of vehicles and the available gaps, but compute the average number of lane-changing frequency based on the speed differences between lanes. This is grounded on the assumption that link traffic tends to evolve to an equilibrium state (Gazis et al., 1961). The following two categories of lane-changing vehicles are identified in the model:

1. Mandatory lane-changing. This refers to vehicles which need to turn at the next intersection but are not in the proper lane to do so. We use the following criteria to characterize mandatory lane-changing vehicles:
    1) vehicles are on urban streets of more than one lane, and
    2) need to turn at the next intersection, and
    3) are within a "decision zone" from the next intersection. We currently set the "decision zone" as one-half of the link length.
2. Discretionary lane-changing. This refers to vehicles that change lanes in order to increase their speeds. For instance, vehicles may change lanes if the speed

differences between the current and neighboring lanes are over some threshold. Since this model simulates individual vehicles, certain driver attributes, such as risk aversion, can be incorporated in the lane-changing mechanism.

Given a randomly assigned threshold for the speed indifference for each driver, the following procedures are used to characterize discretionary lane-changing vehicles:

1) vehicles on urban streets with more than one lane, and
2) need not to turn at the next intersection, or
3) will turn at the next intersection but are currently outside the "decision zone".

The M-MPSM methodology is assigned to model the traffic evolution and speed distribution across lanes in moderate traffic networks. However, a detailed microscopic simulation method should be used under scenarios (such as work zone, lane closure, or congested links) where interactions and frictions between traffic vary substantially even in the same lane. The basic microscopic simulation logic is presented below:

Microscopic Model for Heavy Congested Urban Streets (MICRO)

Microscopic models consider both the interactions between the leading and following vehicles in the same lane and adjacent lanes based on kinematic principles. Such interactions are generally captured through car-following and lane-changing mechanisms.

Car-Following

Various car-following mechanism has been well discussed in the literature (Gazis et al., 1961; Newell, 1961; Lee 1966; Gipps, 1981). Most of them are based on the anti-collision concept. This concept assumes that if two vehicles are in a leader-follower relationship, the follower should be in a safe condition when the leader suddenly decelerates to a stop.

The microscopic simulation methodology presented in this study employs the car-following model developed by Gipps (1981). It is derived by setting limits on the performance of driver and vehicle and using these limits to calculate a safe speed with respect to the preceding vehicle. There are two constraints on the speed of vehicle n at time t + τ. The first one is based on the assumption that vehicle n will not exceed its driver's desired speed. It has the following form:

$$v_{dn}(t+\tau) = v_n(t) + 2.5a_n\tau\left(1-\frac{v_n(t)}{V_n}\right) \times \left(0.025 + \frac{v_n(t)}{V_n}\right)^{\frac{1}{2}}$$

The second speed limitation, based on the anti-collision concept, has the following form:

$$v_{sn}(t+\tau) = b_n\tau + \left(b_n^2\tau^2 - b_n[2[x_{n-1}(t) - s_{n-1} - x_n(t)] - v_n(t)\tau - \frac{v_{n-1}(t)^2}{b_{n-1}}\right)^{\frac{1}{2}}$$

where

a_n = the maximum acceleration for the driver of vehicle n,
b_n = the most severe braking acceptable by the driver of vehicle n ($b_n<0$),
s_n = the effective size of vehicle n (physical length + safety margin),
V_n = the desired speed of the driver of vehicle n,
$x_n(t)$ = the location of the front of vehicle n at time t,
$v_n(t)$ = the speed of vehicle n at time t, and
v_{dn} = the speed of vehicle n according to the desired speed limitation,
v_{sn} = the speed of vehicle n according to the safety limitation,
τ = the apparent reaction time, a constant for all vehicles.

Thus the new speed of vehicle n at time t+τ is given by the minimum between the two speed limits, v_{dn} and v_{sn}:

$$v_n(t+\tau) = \min(v_{dn}(t+\tau),\ v_{sn}(t+\tau))$$

Lane-Changing

The MICRO model uses the same categorization of lane-changing vehicles as the M-MPSM model. However, its lane-change process is more realistic since actual positions of surrounding vehicles in adjacent lanes are checked before the vehicle changes lane.

2. PARALLEL SIMULATION MODEL

In this section, we will present a real-time simulation model implemented on the Connection Machine CM-2 -

a representative member of the SIMD family (Thinking Machine, 1991).

Mapping Traffic Systems on The Connection Machine

There are three principal components in an actual traffic system: (1) traffic network such as urban streets and freeways (2) network control, i.e., traffic signal controls and ramp metering, and (3) traffic or vehicles. In a traffic system, most activities, such as vehicles movements or traffic signal operations, are parallel in nature. By configuring these components as parallel variables, we can simulate the parallel nature of traffic systems.

For each simulation methodology, there are many ways to design these parallel variables. Each configuration will have different advantages and weaknesses in terms of processor and memory requirements, communication patterns, etc. Furthermore, while one configuration works well on one simulation methodology, the same configuration may not work well on the other.

Several design concepts have been analyzed. The proposed design shapes the three main parallel variables (link, node, and vehicle) as a one-dimensional parallel variable. A description of each parallel data structure is given below:

Link parallel variable (LPVAR)

The LPVAR is a one-dimensional parallel variable. Each element of the LPVAR corresponds to a uni-direction link in the network. It contains link information that will be used in each simulation methodology. These information can be divided into two groups. The first group is common for all three simulation methodologies. They includes: link type, link length, number of lanes, speed limit, jam concentration, and link exit control. The second group relates to performance characteristic of each link which are used chiefly by the MPSM and M-MPSM methodologies. They include minimum flow speed and parameters for macroscopic speed-concentration function. The M-MPSM methodology requires these information separately for each lane as opposed to only one link value for the MPSM methodology. The MICRO methodology relies on the interactions between vehicles to compute speed thus only requires LPVAR for path processing function.

Node parallel variable (NPVAR)

The NPVAR is also shaped as an one-dimension parallel variable. Each element represents an intersection between surface streets or an entrance/exit on freeways.

Table 1: LPVAR Information in Each Processor

Common Information
- **Link ID no.**
- **Link length (m)**
- **Number of lanes**
- **Link jam concentration (# of veh.)**
- **Start node #**
- **End node #**
- **Maximum flow speed (m/s)**
- **Start of green (offset, sec)**
- **Start of red (offset, sec)**

MPSM
- **Link parameter for macroscopic speed-concentration equation**
- **Current # of vehicle in a link**
- **Minimum flow speed in a link (m/s)**

M-MPSM
- **Lane parameter for macroscopic speed-concentration equation**
- **Current # of vehicle in each lane**
- **Minimum flow speed in each lane (m/s)**

Whereas some information in LPVAR may be specific to each simulation methodology, all the information in NPVAR are the same for all three simulation methodologies. These information can be divided into two groups. The first group consists of indices of links entering and exiting each node, these indices are used by vehicles for their path processing operation. The second group is related to signal and ramp control. It contains information such as the cycle length, and phase duration. Both signals and ramp metering rate are modeled in the same fashion. The actual ramp meter is implemented as a simple traffic signal, whose cycle length and green phase duration is predetermined to produce the appropriate average ramp metering rate. In this paper, all signals and ramps are assumed to be two-phased, pre-timed control.

Table 2: NPVAR Information in Each Processor

- **Link Index**
  - **Incoming link**
  - **Exiting link**
- **Link Control**
  - **Cycle length**
  - **Ramp metering**

Vehicle parallel variable (VPVAR)

VPVAR is shaped as a one-dimension parallel variable with the number of positions equals to the maximum number of vehicles to be simulated in the network at any time slice. Each element in the VPVAR keeps track of each vehicle and driver key characteristics which include its path, location, current link, current lane, desired speed, and scheduled departure time. At each time step, the moving distance of a vehicle along a link is based on the time-varying speed and the time increment. The time-varying speed can either be computed with the speed-concentration function or be governed by a car-following mechanism. Once vehicles reach the end of a link, they will move onto the next downstream link in a pre-specified path. Upon arrival at its destination, the vehicles will then be removed from the network.

Each vehicle element represents a vehicle in the network; each link element represents either a surface street, a freeway section, or a ramp; each node element represents either an intersection, or a freeway entry/exit. To execute the program, certain processors will need data from others in order to establish relationships among different data entities. For example, the MPSM methodology requires each link to know the number of vehicles currently in that link in order to compute the prevailing speed. Depending on the arrangement of vehicle elements, one can use the general communication or the grid communication to send information between the vehicle and the link parallel data structure to establish those required relationships.

Table 3: VPVAR Information in Each Processor

Entry time
Vehicle ID
Current link number
Current link position
Path index
Path information
Desired speed
Desired acceleration/deceleration rate
Driver type
ATIS equipment

Parallel Model Logic for the M-MPSM Simulation Methodology

Figure 1 illustrates the logic of the M-MPSM methodology. The logic is similar to the original MPSM model by Junchaya et al. (1992), but extends its capability to model integrated networks and lane changes. Furthermore, both the MPSM and the M-MPSM have been combined so that vehicles on both freeways and urban streets can be simulated together to speed up the execution time.

The proposed method contains only one loop for one simulation time period and three stages of execution. In the first stage, parallel variables for vehicles, links, and intersections are initialized from external files. These external files include: (1) traffic demands generated from an O-D matrix of individual vehicles with pre-determined paths, and (2) network information of nodes, links, and signal control. The second stage involves the main simulation routine which consists of five steps: updating link exit control, counting vehicles and updating link speeds, lane changing maneuver, moving vehicles, and updating vehicles links. A summary of simulation statistics is generated in the last stage.

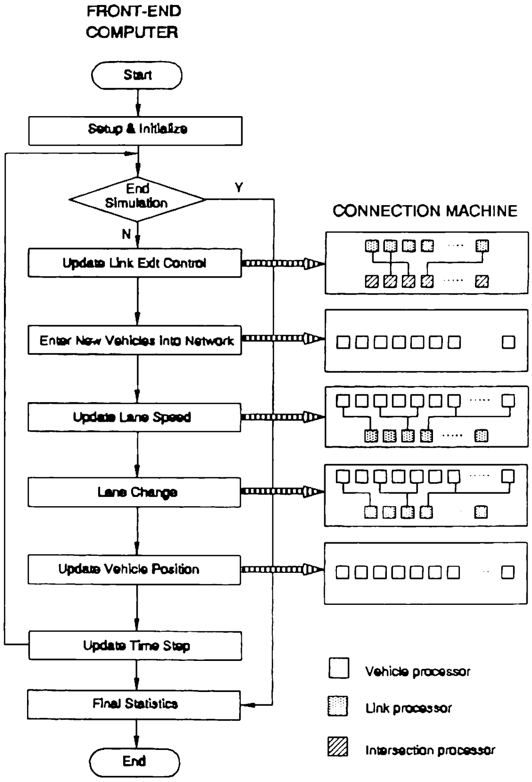

Figure 1: Parallel Model Logic for the MPSM Simulation Model

Parallel Model Logic for the Microscopic Simulation Methodology

The MICRO methodology which use car-following and lane-changing mechanisms can be modeled with the same parallel data structure as in the MPSM and M-MPSM methodology. The main simulation logic for a MICRO model consists of eight modules: updating link exit control, entering network or changing link, computing car-following speed, determining lead-lag vehicles, determining lane-change feasibility, determining lane-changing vehicles, determining target lane, and updating vehicle position (Figure 2). Both the MPSM and M-MPSM methodology utilizes LPVAR to compute the speed in VPVAR via interprocessor communication between the two parallel variables, whereas the MICRO methodology computes the car-following speed and evaluate the lane-change scenarios within the VPVAR itself through a dynamic arrangement of vehicle elements (Figure 3). Thus, some information such as macroscopic link parameters in the LPVAR will not be used in the MICRO methodology. However, the VPVAR needs additional drivers' characteristics such as acceleration/deceleration rate, desired speed, drivers' type, etc.

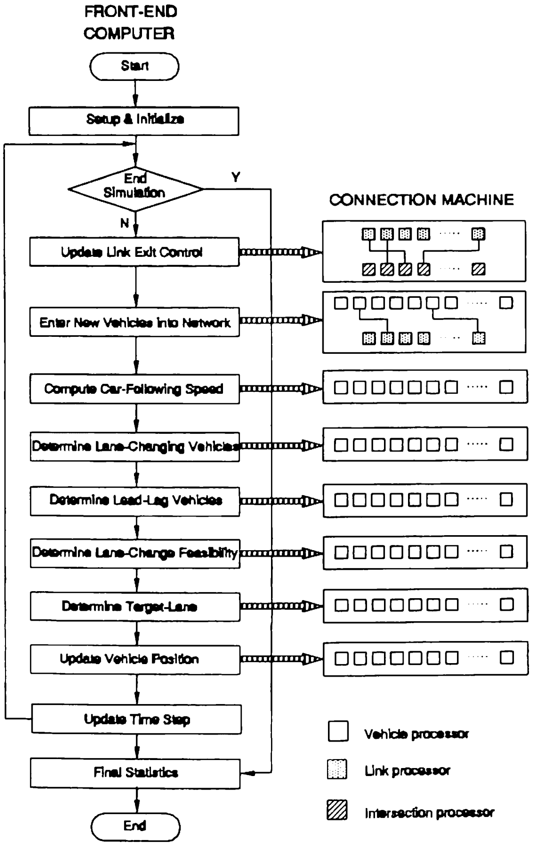

Figure 2: Parallel Model Logic for the Microscopic Simulation Model

3. NUMERICAL EXPERIMENTS

The main purpose of the numerical experiments is to compare the running time for each of the three methodologies under similar environments. Because of the model complexities, it is expected that the MICRO model will take the longest execution time. The original MPSM model should take the least time to complete the simulation since it has the simplest logic among the three proposed models.

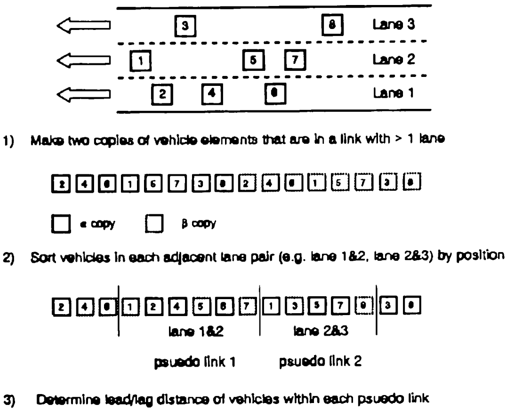

Figure 3: Arrangement of Vehicle Elements for Determining of Neighbor Vehicles

The three simulation methodologies have been implemented on the Connection Machine CM-2 with codes written in C* (Thinking Machine, 1990), an ANSI C standard with parallel extension. This CM-2 has been configured with 16K processing elements running at 7.0 mhz, each processing element having up to 1024 kbits of memory.

The test network is a hypothetical grid network of 100 (10x10) nodes and a total of 360 links. Each two-way street connecting a pair of nodes is represented by two links. The length of each link is assumed to be 1.25 mile. All urban streets for the M-MPSM and the MICRO models have three lanes. The free-flow speed on all links for the MPSM and M-MPSM models is specified as 40 mph with a minimum speed of 5 mph. The jam concentration of each link is set at 120 veh/mile.

The demand level over the 30-minute simulation time period varies from 8,000 to 32,000 vehicles for the MPSM models, and 4,000 to 8,000 vehicles for the MICRO model. The variation in the demand level will result in different virtual processor ratios (VP ratio). In the MPSM model, 8,000 vehicles simulated on the CM with 8K processors will yield a VP ratio of 1:1. However, in the MICRO model, 8,000 vehicles on 8K processors will yield a VP ratio of 2:1, because of the double copies of vehicle elements in the lane-changing computation.

Table 4 and Table 5 reports the execution time of the MPSM and M-MPSM model, respectively. It confirms the finding of the previous paper by Junchaya et al. (1993) about the main factors affecting the performance of the model: number of processors, number of vehicles, VP ratio, and time interval. The execution time of the MICRO model is reported in Table 6.

Table 4: MPSM Model Execution Time Results on the Connection Machine

Number of Vehicles	Time Interval (sec)	Execution Time (sec)	
		8,192 PE*	16,384 PE
8,000	1	74	n.a.
	2	38	n.a.
	4	20	n.a.
	6	14	n.a.
16,000	1	176	98
	2	91	50
	4	47	26
	6	33	18
32,000	1	567	256
	2	281	113
	4	144	67
	6	100	47

*** Processing Element**

As expected, the MPSM execution time is fastest, follow by the M-MPSM and the MICRO model. Because of the lane-chaging mechanism, the M-MPSM model takes longer to complete the simulation as the MPSM model. With 8,192 processors, 8,000 vehicles, and one second time interval, M-MPSM model takes 182 seconds compared to 74 seconds by the MPSM model to complete 30 minutes of simulation. When the number of processors increase to 16,384 processors, simulating 32,000 vehicles using 6 seconds time interval takes the M-MPSM 102 seconds to complete as opposed to 47 seconds for the MPSM.

Table 5: M-MPSM Model Execution Time Results on the Connection Machine

Number of Vehicles	Time Interval (sec)	Execution Time (sec)	
		8,192 PE	16,384 PE
8,000	1	182	n.a.
	2	94	n.a.
	4	49	n.a.
	6	33	n.a.
16,000	1	435	219
	2	221	111
	4	117	58
	6	77	39
32,000	1	1380	573
	2	697	290
	4	353	149
	6	240	102

The logic of the MICRO model is the most complex among the three models presented in this paper involving several interprocessor communication processes. As expected, it takes the longest to complete the simulation. Using 8,192 processors to simulate 4,000 vehicles at time interval of 2 seconds and 1 second for car following and lane change, respectively, it takes 919 seconds to complete 30 minute of simulation. Increase the time interval to 6 seconds will decrease the execution time to a more acceptable range of 284 seconds.

4. CONCLUSIONS

This paper has presented our preliminary network simulation models for real-time operations with a massively parallel SIMD structure. The performance of the parallel MPSM and the M-MPSM models is sufficiently effective for use in a real-time environment. However, the MICRO model does not perform as well as the MPSM or the M-MPSM model.

The authors realize that there are still several research

Table 6: MICRO Model Execution Time Results on the Connection Machine

Number of Vehicles & PE	Time Interval for Lane-Change (sec)	Execution Time (sec)
4,000 Veh. & 8,192 PE	1	919
	2	531
	4	349
	6	284
8,000 Veh. & 16,384 PE	1	1015
	2	586
	4	371
	6	301

issues which have not been fully addressed in the present paper. In particular, we wish to explore other parallel computer architectures such as the MIMD. Many MIMD machines utilize a control-parallelism approach which divides up a standard program into more or less independent subsets of instructions and to assign one such subset to each processor. The main computation issues for the MIMD programming model are synchronization and load balancing among processors. We are currently focusing on developing a network decomposition algorithm that will divide the traffic network for optimal load balancing.

ACKNOWLEDGEMENTS

The contributions of Hani S. Mahmassani and Stavros Zenios are acknowledged. The authors would like to thank Jerry Sobieski of University of Maryland Institute of Advanced Computer Studies (UMIACS) for his valuable inputs and comments. This research is partially supported by FHWA project DTFH 61-90-R-00074 "TRAFFIC MODELLING TO SUPPORT ADVANCED DRIVER INFORMATION SYSTEMS (ADIS)". The authors would like to thank UMIACS, the Massively Parallel Computing Research Laboratory of Sandia National Laboratories and the US Department of Energy for providing computer time on the Connection Machine.

7. REFERENCES

G.L. Chang, H.S. Mahmassani, and R. Herman. A Macroparticle Traffic Simulation Model to Investigate Peak-Period Commuter Decision Dynamics. *Transportation Research Record 1005*, TRB National Research Council, Washington, D.C. , 1985, pp. 107-121.

D.C. Gazis, R. Herman and R.W. Rothery. Non-linear Follow the Leader Models of Traffic Flow. *Operation Research 9*, 1961, pp. 545-567.

P.G. Gipps. A Behavioural Car-Following Model for Computer Simulation. In *Transportation Research 15B*, 1981, pp. 105-111.

T. Junchaya and G.L. Chang. Exploring Real-Time Traffic Simulation with Massively Parallel Computing Architectures. *Transportation Research 1C*, 1993 pp. 57-76.

T. Junchaya, G.L. Chang, and A. Santiago. ATMS: Real-Time Network Traffic Simulation Methodology with a Massively Parallel Computing Architecture. *Transportation Research Record 1358*, TRB National Research Council, Washington, D.C., 1992, pp. 13-21.

G. Lee. A Generalization of Linear Car-Following Theory. *Operation Research 14*, 1966, pp. 595-606.

G.F. Newell. Non-linear Effects in the Dynamics of Car-Following. *Operation Research 9*, 1961, pp. 209-229.

M. Papageorgiou. *Applications of Automatic Control Concepts to Traffic Modeling and Control.* Springer-Verlag, Heidelberg, Germany, 1983, pp. 31.

H.J. Payne. *FREFLO: A Macroscopic Simulation Model of Freeway Traffic-A User's Guide.* ESSCO, San Diego, Calif., 1978,

H.J. Payne. A Critical Review of a Macroscopic Freeway Model, In *Research Directions in Computer Control of Urban Traffic Systems*, ASCE, New York, 1979, pp. 251-265.

Thinking Machines Corporation. *The Connection Machine System, Programming in C** Cambridge, MA. 1990.

AUTHOR BIOGRAPHIES

GANG-LEN CHANG is an Associate Professor in the Department of Civil Engineering at the University of Maryland-College Park. He received a Ph.D. degree from The University of Texas at Austin in 1985. His research interests are focused on developing advanced traffic models for applications in Intelligent Vehicle Highway Systems (IVHS). He is currently involved in adaptive traffic signal control and traffic network incident detection.

THANAVAT JUNCHAYA is a Ph.D. candidate in the Department of Civil Engineering at the University of Maryland-College Park. He received a B.S. in mechanical engineering from Northwestern University in 1982, and he received M.S. in industrial engineering from Northwestern University in 1984. His research interests are focused on traffic models, simulation, traffic operations, and network controls.