

AD-A270 872

STATION PAGE

Form Approved

OMB No. 0704-0188



RT DATE

3. REPORT TYPE AND DATES COVERED
FINAL/15 SEP 90 TO 14 SEP 92

4. TITLE AND SUBTITLE

FIREFLY: A SYNCHRONIZATION
STRATEGY FOR URBAN TRAFFIC CONTROL (U)

5. FUNDING NUMBERS

2304/B3
F49620-90-C-0086

6. AUTHOR(S)

Dr Bernardo Huberman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Xerox Palo Alto Research Center
Palo Alto, CA 943078. PERFORMING ORGANIZATION
REPORT NUMBER

58

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFOSR/NM
110 DUNCAN AVE, SUITE B115
BOLLING AFB DC 20332-000110. SPONSORING / MONITORING
AGENCY REPORT NUMBER

F49620-90-C-0086

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED

12b. DISTRIBUTION CODE

UL

13. ABSTRACT (Maximum 200 words)

The researchers presented a synchronization strategy for urban traffic based on triggering events initiated by traffic lights that improves over existing algorithms. Simulations of urban traffic on a massively parallel machine provide a quantitative measure of the effectiveness of this strategy and allow for comparisons with existing ones. They also showed that augmenting general control procedures with vehicle sensors that provide a measure of queue length in the streets does not lead to improved performance over existing strategies.

DTIC
ELECTE
OCT 18 1993
S A D

14. SUBJECT TERMS

15. NUMBER OF PAGES

21

16. PRICE CODE

17. SECURITY CLASSIFICATION
OF REPORT
UNCLASSIFIED18. SECURITY CLASSIFICATION
OF THIS PAGE
UNCLASSIFIED19. SECURITY CLASSIFICATION
OF ABSTRACT
UNCLASSIFIED20. LIMITATION OF ABSTRACT
SAR(SAME AS REPORT)

93-24349

Firefly: A Synchronization Strategy for Urban Traffic Control

90-C-0086

Baldo Faieta and Bernardo A. Huberman

Dynamics of Computation
Xerox Palo Alto Research Center
Palo Alto, CA 94304

Accession For	
NTIS	CRA&I ☒
DTIC	TAB ☐
Unannounced	☐
Justification	
By	
Distribution	
Availability Codes	
Dist	Avail and/or Special
A-1	

Abstract

We present a synchronization strategy for urban traffic based on triggering events initiated by traffic lights that improves over existing algorithms. Simulations of urban traffic on a massively parallel machine provide a quantitative measure of the effectiveness of this strategy and allow for comparisons with existing ones. We also show that augmenting general control procedures with vehicle sensors that provide a measure of queue length in the streets does not lead to improved performance over existing strategies.

Xerox Rock Center
NE/CRAIG

Final?
Tick?

1. Introduction

Urban traffic, pervasive in modern society, provides a striking example of a distributed system with no global controls. Drivers constantly make local decisions which not always leads to the correct global behavior (i.e. smooth traffic). Moreover, global controls imposed on the traffic lights cannot always accomodate unforeseable changes in drivers behavioral patterns, while having to contend with unavoidable delays in processing information coming from diverse locations. It is therefore of interest to investigate the feasibility of local control algorithms that can lead to improved urban traffic, while at the same time gaining some insights into the operation of distributed control systems.

In this paper we present a synchronization strategy for urban traffic based on triggering events initiated by traffic lights that improves over existing algorithms. We study this new strategy in the context of an urban traffic system made up of traffic signals with their controllers, vehicles with their drivers, and detectors for sensing traffic flow. Moreover, we allow the vehicles to get to their destination by weaving through a network of streets in the presence of other vehicles that have similar behaviors and possibly conflicting goals. Conflicts are resolved to a large extent through a set of rules of behavior (traffic laws) that every driver has to follow. Traffic signals are also present to resolve some of these conflicts in an efficient and safe manner.

Simulations of urban traffic provide a quantitative measure of the effectiveness of this strategy and allow for comparisons with existing ones. We also show that control procedures augmented with vehicle sensors in the streets do not lead to improved performance over existing strategies.

The control of urban traffic by means of signals provides a good benchmark to experiment with distributed and adaptive control schemes. By its very nature, traffic control needs to be decentralized because traffic signals need to react mostly to local traffic conditions. In such schemes, communication between nearby traffic signals is desired in order to provide progression of green lights to vehicles and to coordinate traffic signals to react to unusual events such as fires or gridlocks.

The type of control algorithms one uses in decentralized traffic control is determined by the way the system can sense vehicular traffic. In particular, the use of aggregate measures such as the average speed, the local concentration and the number of vehicles passing through detectors favors simple algorithms that can be easily compared with each other.

The adoption of traffic systems using these adaptive algorithms will depend on how well they can perform in real settings. For this purpose, it is important to assess the average performance and behavior of these envisioned systems through extensive computer simulations.

The system we studied consists of urban traffic in which signal controllers are connected with their neighbors and to vehicle detectors near the traffic signals. Within this system, we have evaluated possible decentralized adaptive schemes for its control.

The simulation was performed on a Connection Machine CM2. The system we studied, with its large number of frequently interacting elements, maps naturally to the massively parallel architecture of the Connection Machine, since each element, i.e. car, traffic light, or sensor, can be assigned to a dedicated processor.

In the following sections, we describe the model of urban traffic we studied, give some general details about its implementation on the Connection Machine and present a new distributed algorithm that we evaluate and compare to existing traffic control schemes.

2. A Model of Urban Traffic

The system we studied consists of vehicles and traffic signals with their corresponding controllers, along with vehicle detectors embedded within a portion of a city. The city being modelled was represented as a network of one way single lane streets arranged in a grid with alternating directions. Street intersections have traffic signals, along with their corresponding controllers that decide what their settings are. These controllers can communicate with other controllers in nearby intersections within the grid. Vehicle detectors are placed in every approach to an intersection and are connected to the controller of that intersection. These detectors can count the number of vehicles crossing them, and can notify the controller when a vehicle has been waiting above the detector pad for more than a certain time.

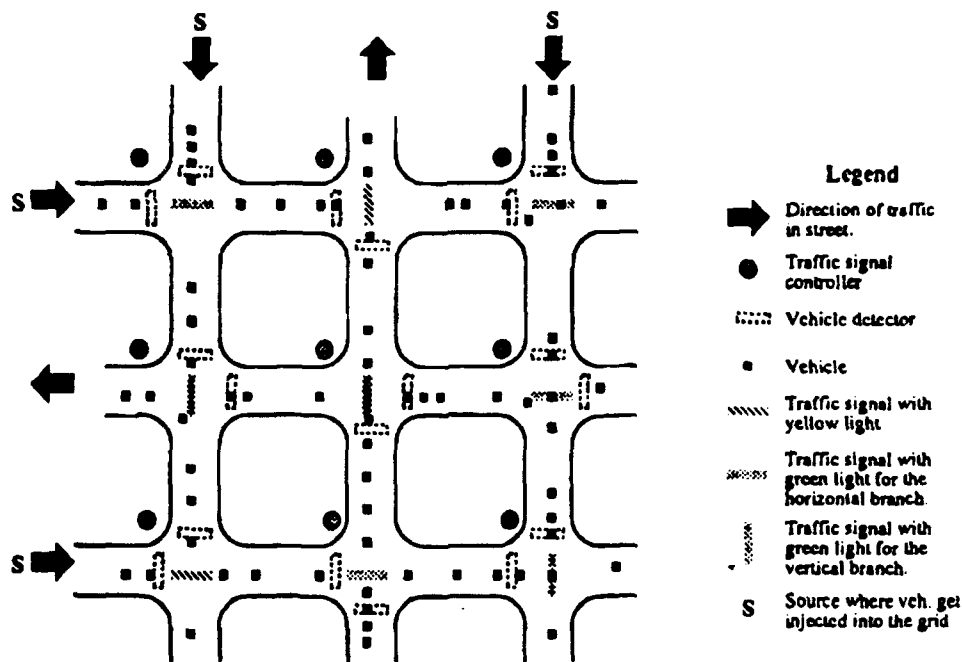


Fig. 1. Shows a representation of a possible state of the system at some time during a simulation experiment. This same representation is used to display an animation of the simulation experiments.

The vehicles in the model were physical objects with inertia and spatial extent. The behavior of the drivers was modelled as part of the behavior of the vehicles. In this sense, one can speak of the modelled vehicles as having intentional behaviors and making decisions.

In the model, most of the vehicles come into the grid through the incoming streets and leave through the outgoing streets within the borders of the grid. Some of them also appear at random locations and times in the streets, thus corresponding to parked vehicles that decide to join the

traffic stream. Vehicles have a final destination which is chosen at random. In order to get there, they attempt to drive through the streets that help them reduce the most the distance between their current location and their destination. As a default, they will only switch directions when either they are moving away from their goal or they have been waiting for too long at the approach of an intersection. Once a vehicle arrives to its destination or whenever it crosses any boundary of the grid, is no longer included as part of the system.

In addition to behaviors ascribed to navigation within the grid, the vehicles possess a variety of behaviors that ensure the proper following of other vehicles and the strict compliance to traffic laws. Thus, vehicles accelerate or decelerate depending on their spacing with respect to the vehicles directly ahead, in a way similar to that described in [16] and [4]. When there are no vehicles preceding them, they maintain a cruising speed as long as there is a green light ahead. If the traffic signal ahead indicates an amber or red light, the vehicles try to stop before entering the intersection to avoid blocking the stream of vehicles crossing perpendicularly. Whenever they find themselves ready to cross an intersection, they decide whether or not to turn and adjust their speed according to that of the vehicles that are going to be ahead, if any, and so as to stay under a maximum turning speed.

The traffic signals in the model are such that whenever a traffic stream in an intersection encounters either an amber or a green light, the other traffic stream in that intersection faces a red light. In this way, local controllers have to decide only which traffic stream should have the right of way. The amber light lasts for 3 seconds and it is set whenever it has been decided that the right of way should be switched. The total cycle time of a traffic signal is twice the amber period plus twice the green period.

Vehicles, traffic signal controllers and vehicle detectors update their state periodically, once every second. However, this updating is done asynchronously in the sense that sometime within the interval of one second, each element of the model updates its state so that all of the elements have updated their state at the end of the interval.

3. Comparing strategies

We have studied and evaluated the performance of several existing strategies, along with a novel synchronization one, inspired by the behavior of coupled oscillators. In order to compare their performance one has to consider methods of evaluation that allow for quantitative measures of traffic flow as a function of speed and density.

There exist different aggregate measures that can be used to compare the effectiveness of a traffic control strategy [9]. Among the aggregates that we can use to compare the different control strategies are: the average vehicle trip and running time, respectively, per unit distance, the average global delay, the average flow of vehicles through every street of the network, the average instantaneous speed and the percentage of vehicles stopped at any time in the network of streets.

In [6], T_m and n are two parameters used to represent the quality of traffic in a network, where T_m denotes the minimum trip time per unit distance. These parameters are derived from the two-fluid model of town traffic [7] using the assumption that the logarithms of T and T_r are linearly related, where T and T_r are the average vehicle trip and running time, respectively, per unit distance. In our model, we record the trip and running times for every vehicle that reaches its destination or exits the grid. From this data we derive T and T_r .

In [13], the authors compare different traffic control strategies using the parameters T_m and n . The vehicle concentrations used in their experiments were below 60 vehicles/mile. With these vehicle concentrations the two-fluid model still holds. However, in our experiments, we are using very high concentrations where we expect the two-fluid to break down [14].

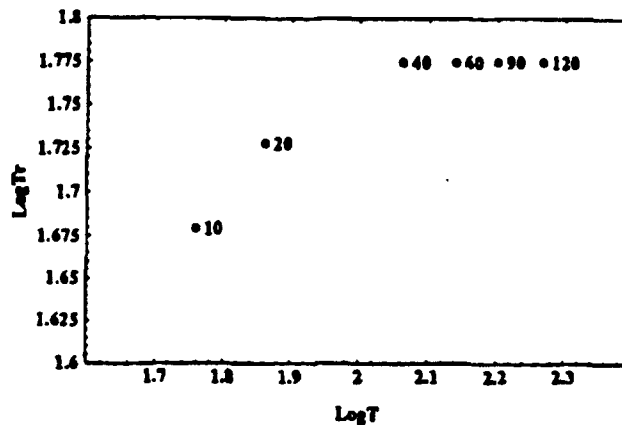
Log-Log plot of T and T_r in min/mi.

Fig. 2. The numbers next to each point correspond to the vehicle concentrations in veh/mi. The aggregates for all points were measured using the random-offsets strategy. The cycle times used in the simulation runs were 18, 18, 30, 40, 40 and 40 seconds corresponding to the concentrations of 10, 20, 40, 60, 90 and 120 veh/mi respectively. The measurements were done over a period corresponding to 10 minutes of running time after having discarded the first 10 minutes.

A Log-Log plot of values of T and T_r measured in our simulation using the random-offsets strategy where the vehicle concentrations ranged from 10 to 120 vehicles/mile shows how the logarithms of T and T_r do not seem to be linearly related. See figure 2. For this reason we have decided not to compare the different control strategies using the parameters T_m and n .

In [13] simulation experiments, once a vehicle has been injected into the system, it stays there as long as the simulation lasts. In their system, the measuring of T and T_r is straight forward since at the end of the simulation they can record these values from every vehicle. Our model assumes a more open system where vehicles are all the time going in and out of the system. Because of this assumption, the system can take much longer to reach a steady state. In order to decide how long should the simulation runs should last, we plot the aggregate branch queue lengths over a period of 20 minutes using the random-offsets strategy at vehicle concentrations ranging from 10 to 120 vehicles/mile.

Aggregate branch queue lengths.

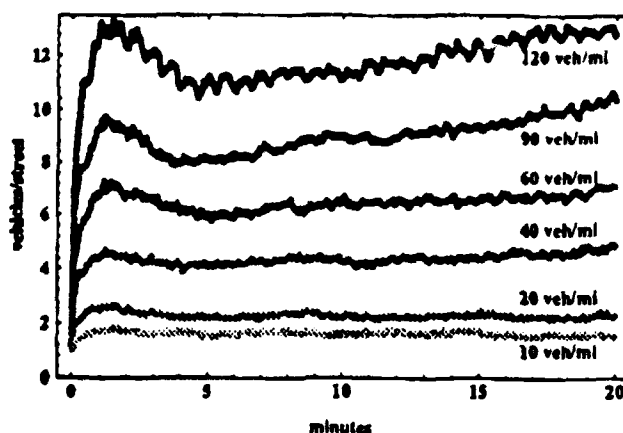


Fig. 3. The aggregates for all the curves were measured using the random-offsets strategy. The cycle times used in the simulation runs were 18, 18, 30, 40, 40 and 40 seconds corresponding to the concentrations of 10, 20, 40, 60, 90 and 120 veh/mi respectively.

As it can be seen in figure 3, during the period of 20 minutes, the queue lengths of the experiment runs with concentrations over 60 vehicles/mile grow slowly over time. The growth is slow enough so that we believe that the period of 20 minutes is long enough to get representative values of real steady state aggregate values. However, we have decided to do 2 sets of experiment runs. The first set, where the simulation runs last for 10 minutes and the first 2 minutes get discarded in order to compute the aggregate values. This set is used to determine the best parameters that should be used for the particular strategies. The second set of simulation runs last for 20 minutes and the first 10 minutes get discarded. In this set of simulations, the parameters found to be best suited in the first set are used in the control strategies. Aggregate values from the second set are used to compare the different control strategies with each other.

As discussed in [5], the measure of the average delay as presented in [17] at an intersection only works when consecutive traffic signals indicate opposite right of ways. Otherwise, the delay should be accounted only in the intersection that differs from the ones up stream. Since it is a problematic measure, we will not use the average global delay to compare between the different control strategies.

The aggregate measure that we use to compare among the different control strategies is the average instantaneous speed and the percentage of vehicles stopped at anyone time in the network of streets. The average flow of vehicles is equal to the the average instantaneous speed multiplied by the concentration [14]. Hence, once we compute the average instantaneous speed curve, the average flow does not contain any extra information.

4. Experimental Results

The simulation runs

During every interval, we recorded the average instantaneous speed over all vehicles and the number of vehicles stopped in the grid. We also recorded, over all the branches in the grid, the average vehicle arrival rate to a branch, the average queue length, and the number of vehicles passing through a detector in a branch. Once the simulation ends we average these values over all the non-discarded simulation cycles.

The size of the grid used in the simulation runs was 16 by 16 intersections. We have tried experiments runs using grid sizes of 32 by 32 and 64 by 64 but we have found no qualitative differences in the results. The modelled distance between intersections is 344 feet. The maximum average speed for any modelled vehicle is 27 mph. The vehicle length is 9.8 feet. Every experiment run begins with the desired vehicle concentration where vehicles are placed at random in the grid. A constant concentration is kept during a simulation run by injecting vehicles at random streets in the borders of the grid every time there are not enough vehicles to make up the desired concentration.

As we mentioned earlier, we have run 2 sets of experiments. In the first set, we do several runs that last for 600 cycles. Since every simulation cycle corresponds to 1 second interval, the runs represent aggregates of 10 minutes observations. The data from the first 2 minutes of the 10 minutes were discarded, and the remaining data from the 8 minutes were used to compute the aggregates. We use the simulation runs in the first set to determine the parameters in which the aggregate average speeds of vehicles, while using the particular control strategies, are the highest. For a particular control strategy and a particular parameter setting, we run the simulation at different vehicle concentrations and measure the aggregates. We then, determine the best parameter settings for a particular control strategy and a given concentration.

In the second set of experiments, the runs last for an equivalent of 20 minutes and the first 10 minutes get discarded for the computation of the aggregates. Using the parameter settings for a particular control strategy found in the first set of simulation runs, we run the simulation at different concentrations for an equivalent of 20 minutes.

The random offsets strategy

In order to evaluate the effects on the overall system of changing the green period duration, we first tested the random-offsets strategy of control, whereby traffic signals have the same green period duration, but the initial offsets at which they start the signal cycles are chosen at random.

We would like to determine the best cycle time settings while using this strategy. We setup the traffic signals in the simulation in a way that correspond to the random-offsets strategy. The traffic signals cycle times that we use in these simulation runs are 18 , 30 , 40 and 50 seconds. These cycle times are tested at vehicle concentrations varying from 10 veh/mi to 120 veh/mi. Figure 4 shows the average speeds and fraction of vehicles stopped while using different cycle times over the range of vehicle concentrations.

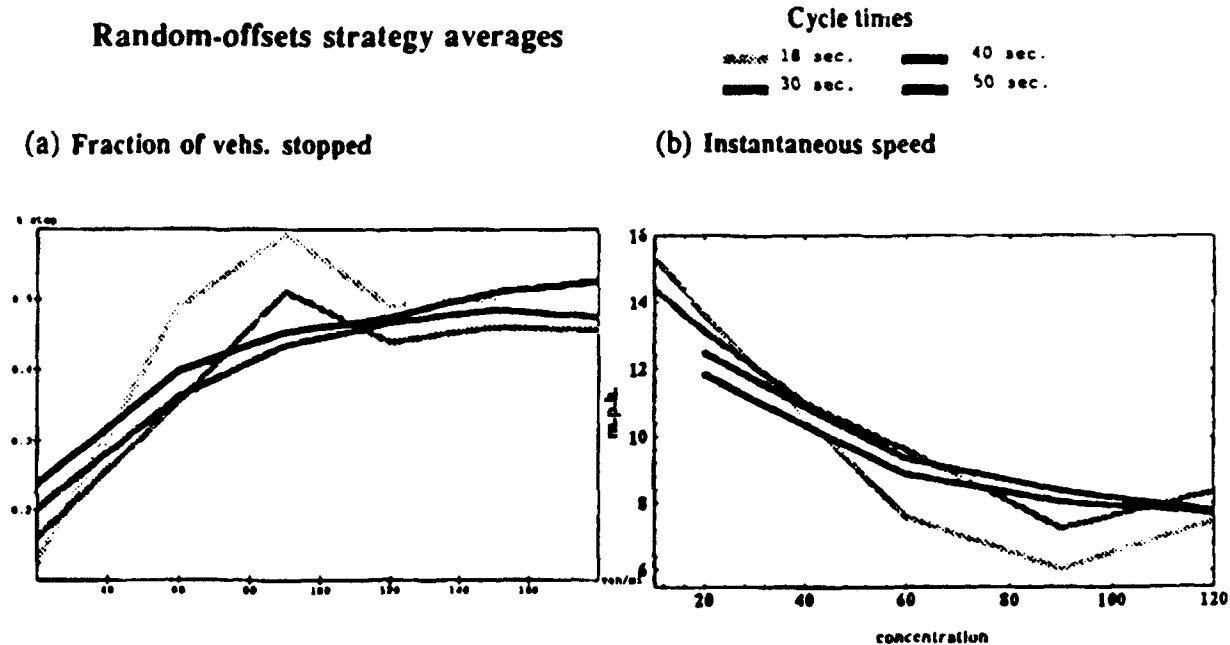


Fig. 4. The plots show the vehicle averages of running the simulation using the random-offsets strategy with the number of vehicles in the grid equivalent to concentrations of 20, 60, 90, 120 vehicles per mile. The plots show different curves with different gray levels. The curves with the lightest gray level corresponding to 18 sec. of cycle time for the lightest curves to 50 sec. for the darkest and 30 and 40 sec. for the curves with gray levels in between. Plot (a) corresponds to the fraction of vehicles stopped. Plot (b) corresponds to the average instantaneous speed in miles per hour.

From the curves, we derive the cycle time settings that should be used at different vehicle concentrations for the random-offsets strategy. Table 1 shows the cycle times that have the highest average speeds given a vehicle concentration for this strategy.

Best cycle times (in seconds)

control strategies	concentration (in veh./mi.)					
	10	20	40	60	90	120
random-offsets	18	18	30	30	40	30

Table 1. Lists the cycle times at which the random—offsets has the highest average vehicle instantaneous speeds. .

An alternating strategy

Next, we tested the effects of traffic signal coordination with the alternating system strategy. As it is described in [18], the green period duration for the alternating traffic signals was set to the time it takes for a vehicle to travel between successive intersections at some average speed. In this strategy, all the traffic signals switch the right of way at the same time. However, traffic signals in successive intersections alternate the right of way of the traffic streams. Notice that this strategy would be equivalent to a strategy where the traffic signals are synchronized in the main arteries in a more general street layout. However, since the layout of the streets is very regular, we do not need to use any method [12] to come out with the correct phase offsets between traffic signals.

We obtain a similar set of curves to the ones in figure 4 for simulation runs where the traffic signals are set up in a way that correspond to the alternating strategy. We decided against deriving these settings as described for instance in [17], [18] since they make use of the flow in the streets and this flow is in turn dependent in the cycle time settings. The values that we picked for the cycle times to be tested are such that their green times derived from the cycle times correspond to the time that it would take a vehicle to go from one intersection to the next at average speeds of 20.1, 16.7, 13.4, 10.1 and 6.7 miles/hour.

As we did for the random offsets strategy, we derive the cycle time settings that should be used at different vehicle concentrations for the alternating strategy. Table 2 shows the cycle times that have the highest average speeds given a vehicle concentration for the alternating strategies.

Best cycle times (in seconds)

control strategies	concentration (in veh./mi.)					
	10	20	40	60	90	120
alternating	22	22	28	28	28	28

Table 2. Lists the cycle times at which the alternating strategy has the highest average vehicle instantaneous speeds. Period times of 22 and 28 seconds correspond to settings of an average speed of 16.8 and 20.1 mph. respectively between intersections.

A vehicle-actuated strategy

Then, we tested a vehicle-actuated strategy, which uses a linear combination of the exact queue-lengths of the two branches at an intersection to decide if the right of way should be switched or not. As is described in [2], the right of way is switched if the green period duration is greater than a maximum allowable duration or if a linear combination of the number of vehicles of the queue being favored is less or equal to the number of vehicles in the unfavored queue.

To implement this vehicle actuated strategy, we assumed that the controllers knew exactly the length of the queues. This assumption implies that in reality at least a new set of detectors would be needed at the beginning of each branch to count the number of vehicles in the branch queue. However, we can envision a mechanism based on a motion-detector algorithm [10], [1], that would in effect keep track of the approximate queue-lengths of the branches by using only one set of detectors at the intersection approaches.

The way that such mechanism would compute the branches queue lengths is as follows. Every controller would determine the number of vehicles leaving its intersection. This variable would be updated by adding to it, during every decision cycle, the number of vehicles passing through detectors in branches leading to the intersection, and subtracting from it the number of vehicles passing through detectors in branches leading away from the intersection. When there would not be a change of this variable, its value would decay to 0 over time. The approximate queue length of a branch would be then a percentage of the number of vehicles leaving the intersection that the branch leads away from. This percentage would be precomputed from an approximate average number of vehicles that turn in the intersection towards the particular branch.

We would like to determine the best settings to be used while using the vehicle-actuated strategy at different vehicle concentrations. As is described in [2], the right of way in a traffic signal is switched if a linear combination of the number of vehicles n of the queue being favored is less or equal to the number of vehicles in the unfavored queue. We express this linear combination as $\alpha n + \beta$ and we run several simulation runs with different α and β values. The α values that we use are 1, 1.5, 2, 3 and 4.5. The β values that we use are 1.5, 3, 6 and 9. The settings where we get the highest average speeds are shown in table 3

Best α and β parameters for the vehicle-actuated strategy

<i>control parameters</i>	<i>concentration (in veh./mi.)</i>					
	10	20	40	60	90	120
α	2	2	3	3	3	3
β	1.5	1.5	6	6	9	9

Table 3. The α values that we use are 1, 1.5, 2, 3 and 4.5. The β values that we use are 1.5, 3, 6 and 9.

Firefly: a synchronizing traffic control system

We now describe a synchronizing strategy which was inspired by the behavior of oscillations in biological organisms. Swarms of fireflies that can flash in synchrony, crickets chirpin in unison, and epileptic seizures provide examples of triggering events leading to synchronization behavior that are common in the natural world. For recent studies, see [15], [3], [11], and references therein.

We now describe the *Firefly* strategy. Initially, the traffic signals in the system are initialized as in the random-offsets strategy. Then, every time a controller switches the right of way in its intersection, a small number (relative to the cycle time) gets added to the current green period duration of the intersections connected to the branches leading away from this intersection. The sign of the small number is such that after adding it to the current green time of the intersection, the traffic signals in these intersections are closer to be coordinated like in the alternating strategy. The best cycle times that we use are the same as the ones used in the random-offsets strategy.

This mechanism has the desired effect that after a short amount of time the light cycles get nearly synchronized. This can be seen from the plots in figure 5 showing the evolution of a row of lights through time in the simulation. In the plots, when the signal corresponding to a light is high, the light is red, otherwise is green. The first plot shows the red/green signals for the first 200 steps in the simulation. The second plot shows the last 200 steps in the simulation. As it can be seen in the second plot, the lights begin to switch from red to green almost at the same time and with the pattern of the alternating strategy.

Red-green light signals for a street

Initial 200 steps

Final 200 Steps

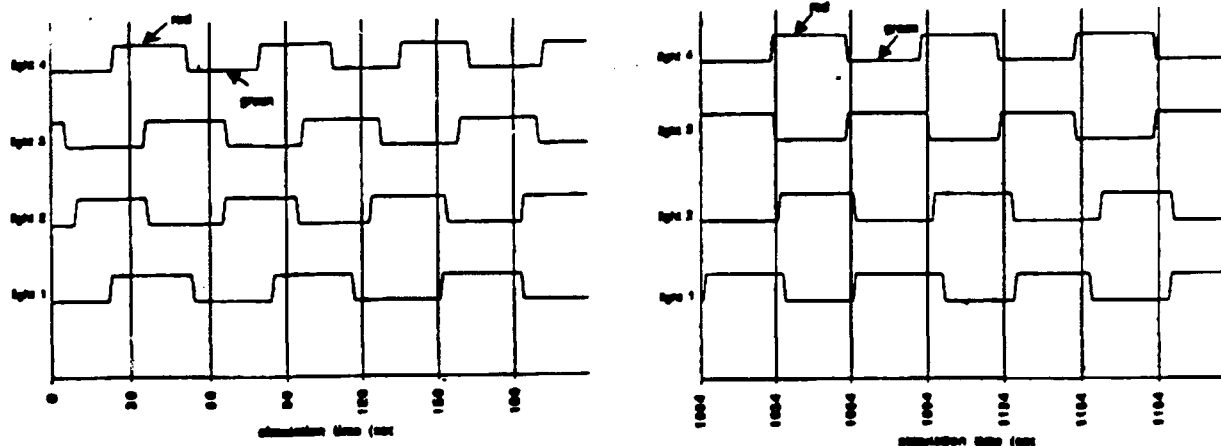


Fig. 5. Red/Green signals for 4 traffic lights in the same street. When the signal is up, it corresponds to red, when the signal is down it correspond to green. the plot on the left shows the red/green signals for the first 200 steps in the simulation. The plot on the right shows the last 200 steps in the simulation

Performance comparisons

The first set of experiments were used to get the parameters for every strategy for which you get the highest performance of the system using that particular strategy.

In the second set of experiments, we use the best parameter settings found in the first set and run the simulation for 20 minutes. To compute the aggregate average speeds, we discard the first 10 minutes of the simulation runs and compute the averages with the left over 10 minutes. As we can see in the figure 6 the synchronizing strategy performs better than the other strategies even though it does not know anything about the queue sizes in the street branches. It seems to be generating the right pattern of delays so that at any concentration it performs better than the other strategies.

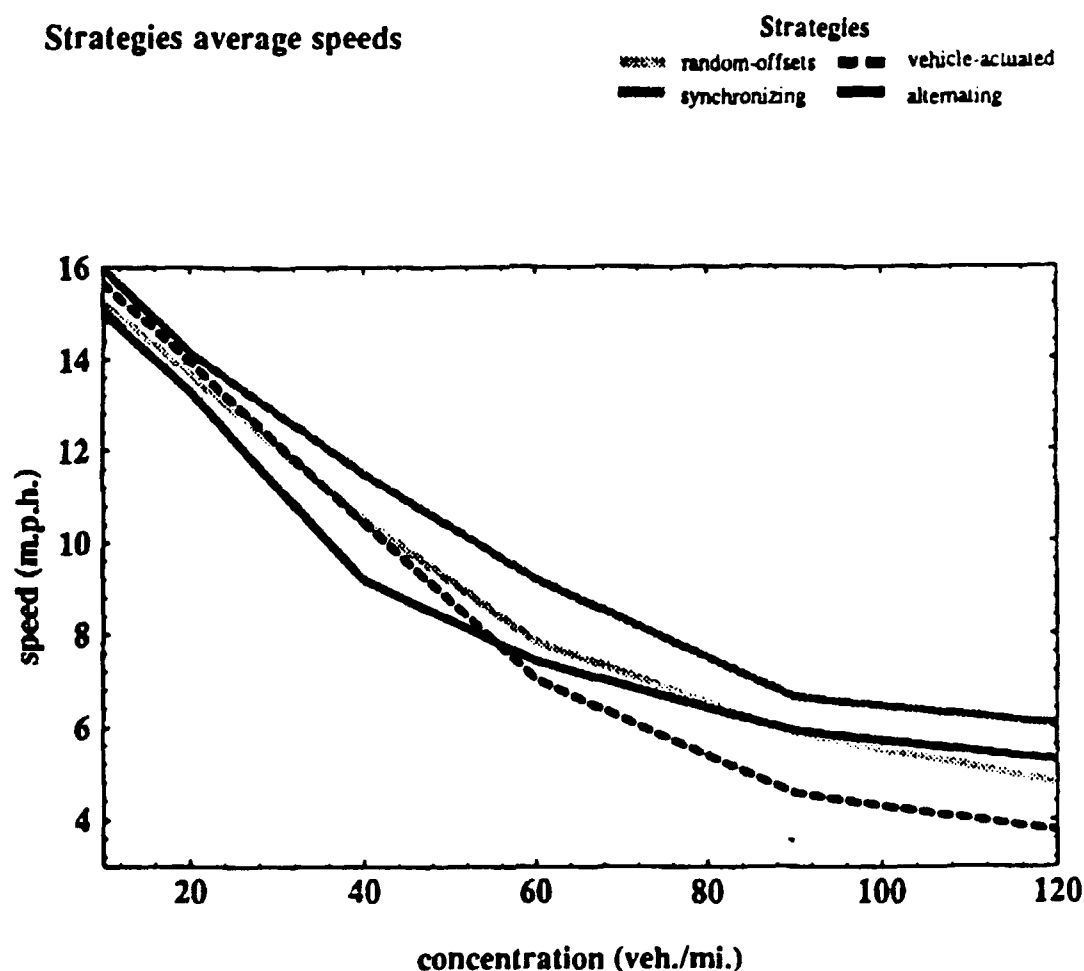


Fig. 6. The plots show the different strategies aggregate average speeds for concentrations ranging from 10 to 120 vehicles/mile.

In order to see whether or not this improved performance is due to small offsets in the synchronization pattern, we performed an experiment in which we added a random fluctuation of controlled magnitude to the synchronization timing of the fully synchronized lights. In this fashion we tested how small uncorrelated offsets affect the performance. As the Figure 7 shows, although such randomization leads to an improvement over all the other control strategies, it fails to account for all of the improved performance observed.

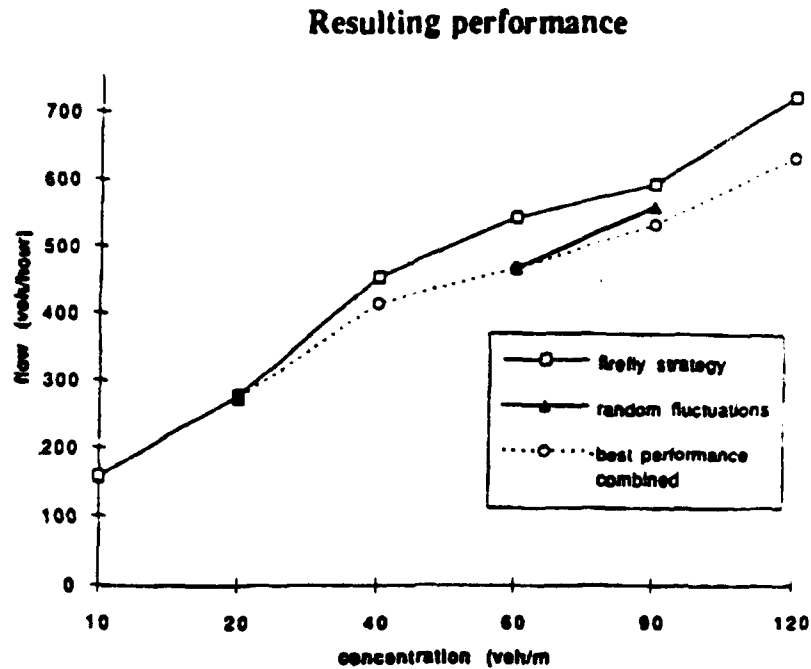


Fig. 7. Flow versus vehicle concentration for the firefly strategy against that obtained by adding random fluctuations..

Thus it seem that the overall performance of this new strategy is due to the correlated part of the small synchronization offsets.

5. Discussion

In this paper we presented a synchronization strategy for urban traffic that performs better than existing control schemes. The basic idea relies on a set of triggering events on traffic lights that lead to increase traffic flow. A number of detailed computer simulations on a massively parallel machine were conducted in order to assess its performance and to compare it to existing algorithms.

The control of urban traffic by means of signals provides a good benchmark to experiment with distributed and adaptive control schemes. The type of control algorithms one uses in decentralized traffic control is determined by the way the system can sense vehicular traffic. In particular, the use of aggregate measures such as the average speed, the local concentration and the number of vehicles passing through detectors favors simple algorithms that can be easily compared with each other.

From our studies we learned that aggregate variables that can be measured within an urban traffic system do not carry enough information for signal controllers to make decisions that affect significantly to the performance of the system. As a result, proposed new systems where vehicles broadcast a variety of data to the system which in turn broadcast some data back, promise to have a better chance at improving the performance of the system.

6. Appendix — Simulation details

The simulation of the model described above was implemented in a CM-2 connection machine [8]. The connection machine is a massively parallel computer with 16384 processors. A program running in the front-end computer broadcasts at every step instructions to a selected group of processors. Each selected processor then executes the broadcasted instruction using the values in its local memory or the values in the memory of other processors. Every processor has an address which other processors will use to communicate with it. Processor addresses are organized either as locations within a grid or as a range of integers.

There is a way to run with multiple virtual processor sets in the Connection Machine . A virtual processor set includes the address organization together with the contents of the local memories. A program running multiple virtual processor sets can be executing instructions in one processor set, switch to another set, execute some instructions in this set and switch back. In this way, a program can have different types of objects processed in different processor sets.

In the present simulation, we use 2 processor sets. The intersections processor set and the vehicles processor set. The first processor set handles all the operations concerning the layout of the streets and the control of the traffic signals. The vehicles processor set handles all the operations that update the vehicles and the queues within streets.

In the intersections processor set, every processor manages what is happening within an intersection, which includes updating the traffic signal controllers and the setting of the traffic signals. The addresses of the processors are organized in a grid that matches the physical layout of the streets so as to profit from the faster communication rates that this organization enables. Indeed, with this mapping neighbor intersections are handled by processors having nearby addresses within the address grid.

The simulation program updates the data structures that correspond to the sensors every cycle. Then it updates the data structures corresponding to the vehicles and the ones that correspond to the traffic signal controllers. The updating of the vehicles and the signal controllers needs to be done asynchronously. To emulate asynchronicity, the program selects at random four groups of processors to be updated in sequence, all within one simulation cycle. Then, it selects at random within a processor which should be updated first, the signal controller or the vehicle managed by this processor.

Each processor within the vehicles processor set keeps the data structures for a single vehicle in the simulation. All the addresses of the processors managing vehicles within the simulation are contiguous. Also, the data structures for vehicles that are behind other vehicles are kept in processors whose addresses have the same ordering. It is as if all the vehicle data structures are

in a single big list of processors with a list element per processor. Then, the list is subdivided in segments. Vehicles within a segment belong to the same street and their addresses hold the same ordering that they have in the street. In this way it is very easy to find out the address of the processor that holds the data for the vehicle ahead, since that address is just one plus the address of the current processor. Queueing and dequeuing is done in parallel by determining for every element in the list the address of their absolute destination. That way, all the elements are moved to their destination in one single parallel operation.

The connection machine can form aggregates of any value stored in every processor. In this way during every cycle, a variety of aggregate variables get recorded in a file.

After all the vehicles and lights have been updated, an image of the city with the vehicles in it is generated and saved. Once a simulation run has finished, the saved images can be played back at a rate of 10 frames per second.

7. Acknowledgements

This research was partially supported by the U.S. Air Force contract No. F49620-90-C-0086. We thank Erik Lumer for useful suggestions and discussions.

References

- [1] H. A. Ceccato and M. Schreckenberg. Motion detection in noisy environments: Phase transitions in a model system. *Physical Review Letters*, 63(19):2152–2155, 1989.
- [2] M. C. Dunne and R. B. Potts. Algorithm for traffic control. *Opns. Res.*, pages 870–881, 1964.
- [3] G. B. Ermentrout. Stable periodic solutions to discrete and continuum arrays of weakly coupled nonlinear oscillators. *SIAM Journal of Applied Mathematics.*, 52(6):1665–1687, 1992.
- [4] D. Gazis. *Traffic Science*. Wiley, New York, 1976.
- [5] D. C. Gazis. Optimum control of a system of oversaturated intersections. *Operations Research*, (12):815–831, 1964.
- [6] R. Herman and S. Ardekani. Characterizing traffic conditions in urban areas. *Transportation Science*, 18(2):101–140, 1984.
- [7] R. Herman and I. Prigogine. A two-fluid approach to town traffic. *Science*, 204:148–151, Apr. 1979.
- [8] D. W. Hillis. *The Connection Machine*. MIT Press, 1985.
- [9] G. Improta. Mathematical programming methods for urban network control. In *Flow Control of Congested Networks*, volume F38 of *NATO ASI*. Springer-Verlag, 1987.
- [10] W. P. Keirstead and B. A. Huberman. Collective detection of motion in the presence of noise. *Physical Review Letters*, 56(10):1094–1097, 1986.
- [11] Y. Kuramoto. Collective synchronization of pulse-coupled oscillators and excitable units. *Physica D*, 50:15–30, 1991.
- [12] J. D. C. Little. The synchronization of traffic signals by mixed-integer linear programming. *Operations Research*, 14:568–594, 1966.
- [13] H. S. Mahmassani, R. Jayakrishnan, and R. Herman. Network traffic flow theory: Microscopic simulation experiments on supercomputers. *Transpn. Res.-A*, 24A(2):149–162, 1990.
- [14] H. S. Mahmassani, J. C. Williams, and R. Herman. Performance of urban traffic networks. In N. H. Gartner and H. H. M. Wilson, editors, *Transportation and Traffic Theory*. Elsevier, 1987.
- [15] S. H. Matthews, P. C. an Strogatz. Phase diagram for the collective behavior of limit-cycle oscillators. *Physical Review Letters*, 65(14):1701–1704, 1990.
- [16] G. F. Newell. Nonlinear effects in the dynamics of car following. *Opns. Res.*, pages 209–229, 1960.
- [17] F. V. Webster. Traffic signals settings. Technical Report No. 39, Road Research Laboratory, 1958.

- [18] F. V. Webster and B. M. Cobbe. Traffic signals. Technical Report No. 56, Road Research Laboratory, 1966.