# The Areawide Real-Time Traffic Control (ARTC) System : A New Traffic Control Concept

Junguk L. Kim, Jyh-Charn S. Liu and Prabaharan I. Swarnam
Department of Computer Science
Texas A&M University

Thomas Urbanik
Texas Transportation Institute
Texas A&M University

## ABSTRACT

This paper presents a traffic control system, called ARTC, which addresses frequent occurrences of congestion and provides areawide traffic progression. The signal controllers in ARTC are interconnected through a computer network. By exchanging traffic flow information among the signal controllers, ARTC provides a new concept in areawide traffic control. With a global view of the traffic in the area, ARTC anticipates congestion. Simulation results of the ARTC prototype control algorithm over a linear road topology is also presented, and the results show significant improvement over an optimized fixed time control. The signal controllers and the computer network are designed to support the real-time communication requirements and a sufficient level of fault-tolerance.

*Index Terms* — Real-time traffic control, traffic congestion, fault-tolerance, computer networks

# 1 INTRODUCTION

The growth of the number of automobiles on the roads has put a higher demand on the traffic control system to efficiently reduce the level of congestion occurrences, which increases travel delay, fuel consumption, and air pollution. Traffic congestion may occur due to exceptional traffic conditions such as peak-hour traffic, accidents, road-side friction, traffic fluctuations, etc.

One of the prevalent problems in current traffic control systems is the inability to respond effectively to transient congestion. One reason is that a major number of the current traffic control systems do not have effective areawide control for a traffic network consisting of freeways, arterials and cross streets. Future directions for research in this approach of traffic control has been pointed out by Haver and Tarnoff [1].

In this paper, we present an efficient areawide traffic control strategy called the *Areawide Real-time Traffic Control* (ARTC) [1] which can provide orderly movement of traffic, shortened average vehicle delay, congestion prevention, and improved road utilization. The signal controllers in ARTC use data from the vehicle detectors placed in all approaches to the intersection under its control. Each controller communicates through a computer

[1] The ARTC is currently under development and this project is supported in part by Texas State Department of Highways and Public Transportation through Texas Transportation Institute under contract number 2-18-89-1245W. The contents of this paper reflect the views of the authors who are responsible for the opinions, findings, and conclusions presented herein. The contents do not necessarily reflect the official views or policies of the Texas State Department of Highways and Public Transportation and does not constitute a standard, specification, or regulation.

The authors Junguk L. Kim, Jyh-Charn S. Liu and Prabaharan I. Swarnam are with the Department of Computer Science and Thomas Urbanik is with the Texas Transportation Institute

network, with its neighboring controllers, to utilize the roads effectively and to prevent congestion due to exceptional conditions. Moreover, regional controllers are introduced to detect network-wide traffic conditions, and provide control that the signal controllers cannot provide, such as non-linear progression, and restriction of traffic influx into congested areas.

In case of a failure of a signal controller in a region, the neighboring controllers and the regional controllers communicate to provide the possible efficient control under those situations. Moreover, the regional controller provides alternative signal plans in these situations which are appropriate to the prevailing traffic. To maintain continued service and efficient performance in the face of failures, signal controllers are designed with a modular organization and redundant hardware.

This paper is organized as follows. In Section 2, we discuss the limitations of the current traffic responsive systems. The logical structure of the ARTC system and the computational algorithms for signal and regional controls are presented in Section 3. We then summarize the results of our simulation performed for a linear road topology, comparing a fixed time signal control, whose timing plans were computed by TRANSYT7F, with the ARTC system in Section 4. Section 5 discusses the structural design of the signal controller and the communication network among the controllers. We discuss our future plans and conclude this paper in Section 6.

## 2 BACKGROUND

The current computer controlled signal systems can be classified into two categories: *fixed* time control which uses signal plans computed off-line using past data, and *traffic-responsive* control which computes the signal plans according to the prevailing traffic flow [2, 3, 4, 5, 6, 7, 8]. There are different methods to compute the signal timing plans off-line, one system, TRANSYT, is popular among these systems for network wide signal timing optimization [9]. Most of the on-line control systems, which are computer controlled, fall under the traffic-responsive category [10, 11, 12, 13, 14, 15].

The TRANSYT system computes the signal timing plans, given the geometry of the traffic network and the average traffic influx at each approach. This system is based on a macroscopic model of the traffic network and uses a weighted average of the delay and stops as the performance index, which is kept at a minimum when designing the cycle and split. Since the signal plans are designed with the average flows, the actual traffic flow fluctuations is not taken into consideration in TRANSYT.

The traffic network under SCOOT is divided into regions and each region is controlled by a regional controller which periodically (A minimum interval period of $2\frac{1}{2}$ minutes) calculates the cycle and split which is varied by small increments or decrements, usually 4 second variations [11]. For each cycle in a region, the regional controller in SCOOT *calculates* the offset for each intersection based on information collected from the signal controllers. The delay time and the number of stops are used as the performance index measure. A few seconds before a signal change at a intersection, the signal controller may extend the current phase time within the cycle length to adapt to a sudden traffic fluctuation.

SCAT also divides the traffic network into regions, and each region is controlled with one common cycle length as in SCOOT. The regional controller in SCAT can change the forthcoming cycle length after the completion of the current cycle. The offset and split plans in SCAT, which are computed by using off-line methods, are provided and *selected* by the controller on the basis of the current traffic flow.

The traffic network in the ATSAC system is also divided into regions, and each region is controlled by a regional controller [16]. There are three options in choosing the signal timing plans: the time-of-day, the traffic flow, and the one made by the traffic engineer who observes the traffic flow. Failed signal controllers are detected and displayed at the traffic center.

In the above systems the signal plans for each intersection are calculated by considering only the level of traffic flowing into the approaches and does not consider the rate at which the traffic can actually flow through at the downstream approaches. This may result in the downstream approach being overloaded if it was not able to accept the flow. Moreover, the signal controllers heavily depend on the regional controller to compute the signal plans, and hence when the regional controller fails, all the signal controllers in the region should fall back to the primitive control strategy of using fixed time plans. Since most of the computations for the signal plans are performed by the regional controller, it is expected to perform these computations for the entire region in a few seconds, and hence there is a high computation overhead on it. Moreover, the communication network at the regional controller may be congested due to enormous influx of messages from the signal controllers when they transmit their detector data to the regional controller, and a much higher reliability of the network is required.

## 3 THE ARTC SYSTEM

The ARTC system, whose logical structure is shown in Figure 1, is a traffic-responsive control system which provides traffic control, based on current traffic information collected from detectors embedded in each lane of an approach.

The signal controllers in the ARTC system are designed to store data which reflects

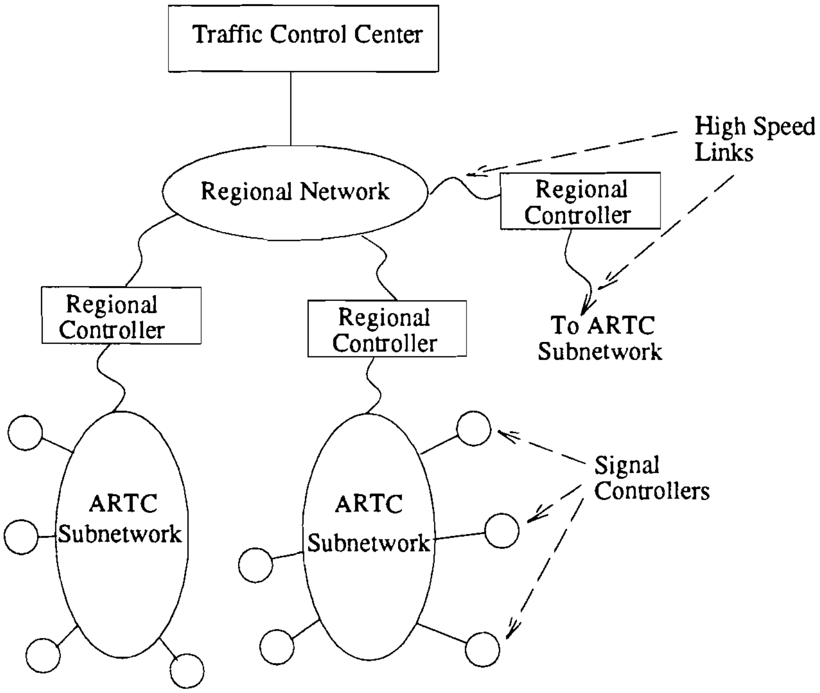

Figure 1: ARTC System Model

the history of the traffic flow collected from detectors, *i.e.* the number of vehicles detected during each unit period of time is stored so that the intersection can estimate an arrival pattern using this data. This type of data storing technique, called Cyclic Flow Profile ($CFP$), is used in SCOOT [17], which approximately reflects the sizes of platoons and their locations in the approach.

In each lane the *entry detectors* and *exit detectors* are placed at the entry point and on the stop line of the lane to keep count of the vehicles entering and exiting the lane respectively. The detectors are assumed to estimate the speed of the vehicles hence the controller can differentiate between light and heavy traffic.

## 3.1 ARTC Strategy

The objective of ARTC is to provide a green duration for an approach such that the queue formed at the intersection is dissolved and the additional vehicles which arrive behind the queue in the approach are also dissipated. Let us say that *Vehicles-Dissolved* is the number of vehicles which pass through an intersection during the green phase. We try to provide a green phase for these vehicles such that it results in an optimal minimum delay per car waiting at the approaches in the red phase. The *Vehicles-Dissolved* consists of two parts; the number of vehicles which are in the queue, denoted by $L_q$ and the number of vehicles which pass the intersection without stopping at the intersection, denoted by *Vehicles-Extension.*

The green duration required for the *Vehicles-Dissolved* to pass the intersection is $t_{green}$ given by,

$$t_{green} = t_{queue} + t_{ext}$$

where $t_{queue}$ is the time required for the queue to dissolve and $t_{ext}$ is the extension time for the *Vehicles-Extension* to pass the intersection without experiencing any delay at the intersection. In the above equation $t_{queue}$ can be broken down into two parts, which are $t_{queue-move}$, the time required for the last car, stopped at the end of the queue, to begin moving and $t_{queue-travel}$, the time required for the last car in the queue to travel to the intersection and exit. To compute these times we need to estimate $L_q$ which can in abstract be defined as follows:

$$L_q = f_1(f_{entry}, t_{red}) \text{ or } f_2(CFP, f_{entry}),$$

where $f_1()$ and $f_2()$ are functions which estimate the queue length using $f_{entry}$ which is the entry flow rate at the approach, $t_{red}$ which is the duration of the red phase during which the queue was formed, and $CFP$ which is the Cyclic Flow Profile data. The queue length can be estimated by using either the function with flow rates which will result in an average queue value over a period of time, or the function with the $CFP$ which will result in estimating the queue by using the information about location of the vehicles present in the approach. These queue estimation methods are explained later in this section.

Once we estimate $L_q$ we can now find the time $t_{queue-move}$ and $t_{ext}$ as follows. As shown in Figure 2a. the initial queue is grouped at the end of the approach, and the time for the last vehicle in this initial queue to begin moving, as shown in Figure 2b, is approximately $(\frac{L_q}{S})$ where $S$ is the saturation flow. The loss time for the vehicles while they accelerate and decelerate can be reflected by appropriately adjusting the flow rate values. We can also note that in Figure 2b, while the initial queue is being discharged, new vehicles may arrive, at a flow rate $f_{entry}$, and append to the queue. Moreover, we can observe that each vehicle in the queue will occupy a space, equivalent to the physical length of a vehicle, and the vehicles following behind the queue, will have to append to this queue. This space occupied by the queue forces the vehicles, appending at the end, to stop earlier than they would, if the queue was not present, hence the rate at which the vehicles stop at the queue is higher than $f_{entry}$. This new rate $f_{pileup}$, at which the vehicles append to the queue, is approximately $\frac{1}{(1/f_{entry}-t_{vehicle})}$, where $t_{vehicle}$ is the time taken for a vehicle to move one effective vehicle-length, which includes the length of the vehicle and the spacing between vehicles while they are stopped in a queue. In order to discharge the vehicles in the initial queue and the new arrivals, we readjust the effective queue discharge rate as $(S - f_{pileup})$ and hence the time for the last vehicle to begin moving is

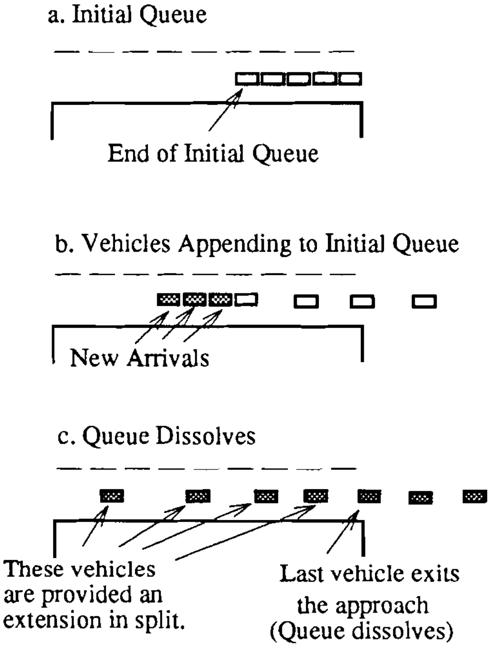
a. Initial Queue
End of Initial Queue
b. Vehicles Appending to Initial Queue
New Arrivals
c. Queue Dissolves
These vehicles are provided an extension in split.
Last vehicle exits the approach (Queue dissolves)

Figure 2: Pattern of Dissolving $L_q$

$$t_{queue-move} = \frac{L_q}{(S-f_{pileup})}.$$

The entire queue is considered to be discharged only when all the stopped vehicles in the queue exit the intersection. Then the approximated time for the last stopped vehicle to exit the intersection would be:

$$t_{queue-travel} = (L_q + f_{pileup} \times \frac{L_q}{(S - f_{pileup})}) \times t_{vehicle}.$$

The time $t_{queue}$ is now given as:

$$t_{queue} = \frac{L_q}{(S - f_{pileup})} + (L_q + f_{pileup} \times \frac{L_q}{(S - f_{pileup})}) \times t_{vehicle}.$$

Now that we have computed $t_{queue}$, we have to find $t_{ext}$. In Figure 3, the height of the triangles illustrated reflects the queue length in an approach. The slopes of the triangle indicate the flow rate at which the vehicles enter the approach and exit the approach. As the queue becomes longer, the delay time for the vehicles increases since it takes a longer time to dissolve the queue, and this delay is proportional to the area of the triangles in Figure 3.

To compute the optimal extension, we first estimate the flow rates for both directions, *i.e.* $f_{red}$ and $f_{green}$, which represent the flow rates at the approaches in the red phase and green phase respectively and then estimate $D_{red}$ and $D_{green}$. Where $D_{green}$ is the delay per vehicle experienced by the vehicles which are in the approach, currently in the green phase, and $D_{red}$ is the delay per vehicle experienced by the vehicles which are in the approach, currently in the red phase. The delay per vehicle during the past cycle, in the approach currently in the green phase would then be:

$$D_{green} = \frac{\Delta_0}{(f_{green} \times (t_{red} + t_{green}))},$$

where $\Delta_i$, where $i = 0, 1, 2, 3, 4$, represents the delay experienced by the vehicles waiting at the queue, as shown in Figure 3. The delay per vehicle during the past cycle, in the approach currently in the red phase would be:

$$D_{red} = \frac{(\Delta_1 + \Delta_2 + \Delta_3)}{(f_{red} \times (t_{red} + t_{green}))}.$$

However, if the current phase is extended by $t_{ext}$ then

$$D_{green} = \frac{\Delta_0}{(f_{green} \times (t_{red} + t_{green} + t_{ext}))}, \text{ and}$$

$$D_{red} = \frac{(\Delta_1 + \Delta_2 + \Delta_4)}{(f_{red} \times (t_{red} + t_{green} + t_{ext}))}.$$

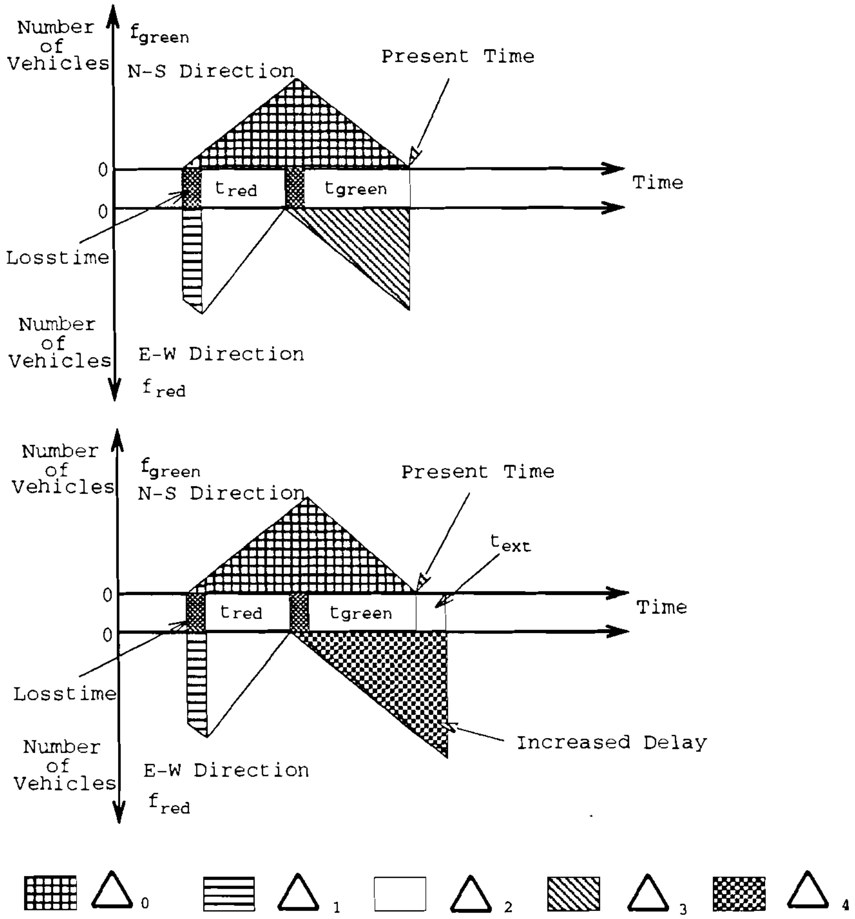

Figure 3: Computation of Extension for Split

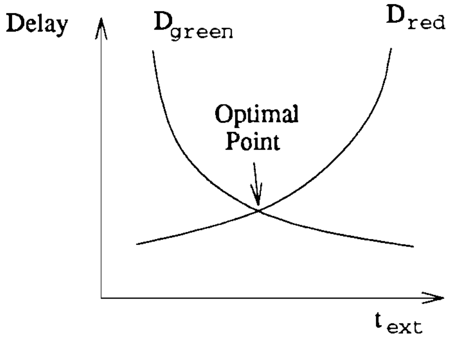

Figure 4: An Illustrative Plot of $D_{green}$ and $D_{red}$ Against Time

We can observe that when the value of $t_{ext}$ is increased for an approach in the green phase, more vehicles pass through the intersection without having to experience any delay due to waiting. From the above control strategy it can be seen that an increase by $t_{ext}$ decreases the delay value in the link which is at green phase. That is, the value of $\Delta_0$ remains same for any value of $t_{ext}$, since the queue has been dissolved and all further vehicles pass the intersection without any delay. Hence, an increase in $t_{ext}$ will result in the the value of $D_{green}$ being reduced. However, an increase in $t_{ext}$ results in the approach in the red phase being extended for the same amount of time which results in an increase in $D_{red}$. If we observe from Figure 4, which shows the anticipated plot of $D_{green}$ and $D_{red}$ against time, an optimal value of the delay experienced by vehicles in both the phases is when both $D_{green}$ and $D_{red}$ have the same value as indicated in the plot.

Moreover, we can represent $D_{green}$ and $D_{red}$ as a total delay per vehicle $D$ as shown below:

$$D = F_1(D_{green}, D_{red})$$

where $F_1()$ is a function which takes $D_{green}$ and $D_{red}$ as parameters. However,

$$D_{green} = g_1(t_{redNS}, t_{redEW}, t_{extNS}, t_{extEW}, f_{red}, f_{green}) \text{ and}$$

$$D_{red} = g_2(t_{redNS}, t_{redEW}, t_{extNS}, t_{extEW}, f_{red}, f_{green}),$$

$$\text{Hence, } D = F_2(t_{redNS}, t_{redEW}, t_{extNS}, t_{extEW}, f_{red}, f_{green}),$$

where $g_1()$, $g_2()$ and $F_2()$ are functions which take, $t_{redNS}$, which is the red time for the North-South direction, $t_{redEW}$, which is the red time for the East-West direction, $t_{extNS}$, which is the extension time for the North-South direction and $t_{extEW}$, which is the extension time for the East-West direction, as parameters and produce the delay times.

Since $D$ is a function of $t_{extNS}$ and $t_{extEW}$, the only unknown parameters, the optimal points $(t^*_{extNS}, t^*_{extEW})$ can thus be obtained by having $\partial D/\partial t_{extNS} = 0$ and $\partial D/\partial t_{extEW} = 0$.

Then by substituting the values of $t^*_{extNS}$ and $t^*_{extEW}$ for $t_{extNS}$ and $t_{extEW}$ respectively, we get the optimal split value.

**Method 1**

In ARTC, each signal controller measures the traffic flow entering each of its approaches during the past few cycles and, the average traffic flow is computed using this information. This method has two main steps; the estimation of the queue using the CFP data and the computation of the split time. We now explain the basic steps in the split computation, which is performed by each signal controller a few seconds before a phase change:

1. Estimate the total number of the vehicles in all the approaches to the intersection and then estimate the size of the corresponding queues at the approaches using the $CFP$ data collected from the detectors.

2. Re-adjust the sizes of the queues to reflect the vehicles which append to the queues while the queue is being dissolved, and pick the maximum queue length value, among the approaches, for which the split has to be computed. Compute the time $t_{queue}$ to dissolve this queue.

3. Using the traffic flow rate during the past cycle, compute the delay experienced by vehicles in both directions, *i.e.* $D_{green}$ and $D_{red}$.

4. Initialize the extension time for the split $t_{ext} = \delta$, where $t_{ext}$ is the time required to extend the green phase and $\delta$ is the minimum time for which a phase can be extended.

5. Repeat the following steps until $D_{green} \leq D_{red}$

   (a) Extend current green phase *i.e.* $t_{ext} = t_{ext} + \delta$.

   (b) Recompute values of $D_{green}$ and $D_{red}$.

6. If $t_{ext} = 0$ then change the phase else extend the current phase by $t_{ext}$.

We now explain the method by which the signal controller estimates the size of the queue, *i.e.* $L_q$, at an approach. For simplicity, we make a few assumptions in our model: the vehicles in the traffic system travel at a uniform speed; the length of all vehicles are equal; and no parking lots or side streets are present in the system.

The vehicles in an approach come to a stop at an intersection, when the facing signal lights for that direction are red, and form a queue which has to be estimated for the split computation. It is difficult to accurately calculate the queue size at an intersection without using expensive vehicle detectors placed throughout the approach; instead factors such as presence of side streets and parking lots, location of vehicles in the approach, variable speeds of vehicles, and different sizes of vehicles, should be considered in the estimation.

The $CFP$ data structure, formed with the data provided by the entry detectors, represents the position of the vehicles in the approach over a period of time, as shown in Figure 5; that is each slot of the $CFP$ in the window reflects the distance from the intersection. The $CFP$ data is updated periodically by shifting the data when new vehicles enter the

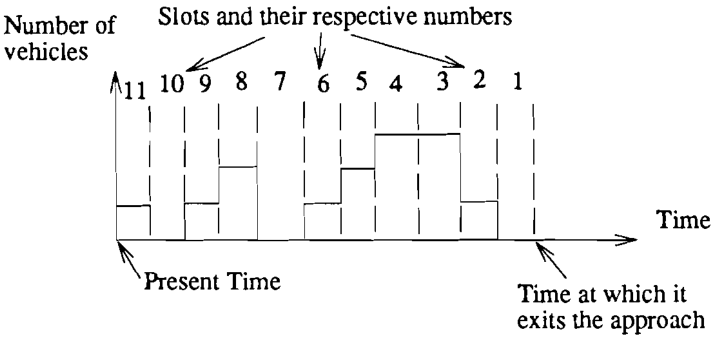
Number of vehicles
Slots and their respective numbers
11 10 9 8 7 6 5 4 3 2 1
Time
Present Time
Time at which it exits the approach

Figure 5: $CFP$ Data Representation

approach, indicated at the leftmost part of the $CFP$ in Figure 5. The vehicles close to the intersection, indicated at the rightmost part of the $CFP$ in Figure 5, get shifted out from the $CFP$ structure when the signal is green. However, if the signal is red, the vehicles will not be able to exit the approach and hence will form a queue. Thus, when the signal is red, the vehicles at the right end of the $CFP$ are shifted into a new data structure called the $QUEUE$, which reflects the number of vehicles guaranteed to be in the queue. When the signal changes to green, vehicles in the queue will exit the approach and hence, as vehicles are detected to exit the approach the $QUEUE$ data structure is decremented.

To estimate the queue in an approach, the flow rates of the left lane, center lane and the right lane of the corresponding approach, are computed using the history data and are represented by $f_{left}, f_{center}$ and $f_{right}$ respectively. For estimating the queue the flow rates which exit each lane are used and the computation is performed during the start of a split in each cycle. Now we explain, in the following steps, how the $QUEUE$ and $CFP$ data structures are used in estimating the queues:

- Compute the minimum queue length of each lane:

  $MIN\text{-}Q_i = (f_i/f_{total}) \times QUEUE$, where $i \in left, right, center$;

and $f_{total} = f_{left} + f_{right} + f_{center}$;

- $APPENDING\text{-}Q_i = MIN\text{-}Q_i$
- Repeat for each lane
  1. Count the vehicles present in the slots of the $CFP$ which would have been occupied by the vehicles in $APPENDING\text{-}Q_i$; the vehicles in these slots, indicated in the $CFP$, should have joined the queue. Let $Num$ denote the count of these vehicles.
  2. $APPENDING\text{-}Q_i = Num \times f_i$
  3. $MIN\text{-}Q_i = MIN\text{-}Q_i + APPENDING\text{-}Q_i$, where $i \in left, right, center$;
- Until $APPENDING\text{-}Q_i < 1$;
- $Q_{Approach} = Max(MIN\text{-}Q_i), i \in left, right, center$;

After estimating the queue by using the above mentioned steps, we compute the split based on the above queue using the optimization for the $t_{ext}$ as explained earlier.

**Method 2**

In this method we use only the average traffic flow values from the detectors in each of the approaches. We now explain below, the steps involved in estimating $L_q$. Let $flow_{entryNS}$ be the maximal entry flow rate for the North-South direction and $flow_{entryEW}$ be the maximal entry flow rate for the East-West direction. Let $\lambda_1$ be the flow rate at which vehicles append to the queues in the North-South direction which is $\lambda_1 = \frac{1}{(1/flow_{entryNS}) - t_{vehicle}}$ and let $\lambda_2$ be the flow rate at which vehicles append to the queues in the East-West which is $\lambda_2 = \frac{1}{(1/flow_{entryEW}) - t_{vehicle}}$. Let $t_{greenNS}$ and $t_{greenEW}$ be the time required for the green phases of the North-South and East-West direction respectively, and let $t_{redNS}$ and $t_{redEW}$ be the duration of the red phase for the North-South direction and the East-West direction, respectively. Let $t_{loss}$ denote the loss time during a phase change and $S$ denote the saturation flow rate. Let $t_{extNS}$ and $t_{extEW}$ be the extension time of the split length of the North-South direction and the East-West direction, respectively. Using the above we have the queues $L_{qNS}$, the queue in the North-South direction and $L_{qEW}$, the queue in the East-West direction which are given as,

$$L_{qNS} = \lambda_1 \times t_{redNS},$$

$$L_{qEW} = \lambda_2 \times t_{redEW}$$

Substituting for $L_q$ in the equation for $t_{queue}$ described earlier we have,

$$t_{queueNS} = \frac{L_{qNS}}{(S-\lambda_1)} + (L_{qNS} + \lambda_1 \times \frac{L_{qNS}}{(S-\lambda_1)}) \times t_{vehicle}$$

Now we have the green times of each approach,

$$t_{greenNS} = t_{queueNS} + t_{extNS}, \tag{1a}$$

$$t_{greenEW} = t_{queueEW} + t_{extEW}, \tag{1b}$$

Since, the red phase for one direction is actually a green phase of the other, in addition to the loss time preceding and following the green phase, we have,

$$t_{redNS} = t_{greenEW} + 2t_{loss}, \tag{2a}$$

$$t_{redEW} = t_{greenNS} + 2t_{loss}, \tag{2b}$$

Thus by using equations (1a), (1b), (2a) and (2b) we can reduce $t_{greenEW}$ and $t_{greenNS}$ to be dependent on the variable parameters $t_{redEW}$, $t_{redNS}$, $t_{extEW}$ and $t_{redNS}$. Now using the optimization for the values of $t_{ext}$ we can find the optimal values of the split and the cycle.

### Progression and Congestion Control

Each signal controller measures the average flow rates of the vehicles entering each of its approaches and if the value of the flow rate in any direction indicates a steady increase and maintains a higher level of traffic when compared to the traffic flow in the crossing direction, then the neighboring signal controllers, along that steady flow, are requested to provide an offset such that a progressive platoon service is provided. This progressive platoon service ensures that the vehicles in the platoon, which are provided such an offset, will pass through the intersection without having to stop at that intersection.

We explain how this progressive platoon service is to be performed for an arterial stretch, which we chose for simplicity. When the traffic flow in an arterial stretch is detected to dominate the traffic flow of the approaches crossing the arterial stretch by the controllers along the path, the following steps are performed.

- The controllers exchange information to identify the path of heavy flow and hence identify the controllers along that path.

- Once the ends of the path is determined, the controllers at both these ends collect the flow and queue information from all the controllers along that path and decide the proposed split duration assuming that the platoon size is constant while traveling through the path.

- This proposed split duration is then propagated to the controllers in that path.

- Each controller estimates the queues formed in the arterial stretch, and appropriately computes the offset between its neighboring controllers such that the queue in the arterial stretch at the intersection is dissolved before the vehicles from the neighboring intersection arrive in that phase.

- Recompute the split duration every few cycles by sending a request to the end controllers.

When there is a heavy flow in both directions of a stretch of road, then it is very difficult to give a progressive service to both traffic flows. However, it is reasonable to give a progressive service to one direction of the traffic if it is extremely higher than the other. If the traffic flowing in both directions, for example east and west, are comparably high, then we provide a partial progressive service for both the flows, which we call the *m:n progression.* That is, the total number of intersections is $(m + n)$ along the path of heavy flow, which is in both directions of the road. Then a ratio of the flows will provide an indication as to which flow gets a progression of up to $m$ intersections with offsets computed to provide a progressive service. This progression of traffic flow and an arterial stretch is shown in Figure 6. This concept can be extended to a non-linear progression in a network.

For example, with a $(m + n)$ intersection linear road topology traffic model, if the ratio of the traffic flow from the east direction to the flow from the west direction is 4:1, then the flow from the east direction will get a progression up to 80% of the intersections and the flow from the west will get a progression up to 20% of the intersections.

Congestion control is enforced by limiting the inflow of vehicles into the approach leading to the neighboring intersection. Before each phase change, the signal controller in each intersection gathers the load values of the approaches leading into its neighboring intersections. If a particular approach carries a load higher than a predefined threshold value, then it limits the influx of vehicles into that approach by appropriately turning the signal lights for that approach to red. Once the affected approach dissipates its load, the neighboring signal controller will resume normal operation and send vehicles into the approach. If the congestion persists beyond a specified period of time, then the signal controllers which lie along the paths leading to that affected approach are informed about the calamity. These controllers are requested to shorten their split length, thus reducing the overall influx into

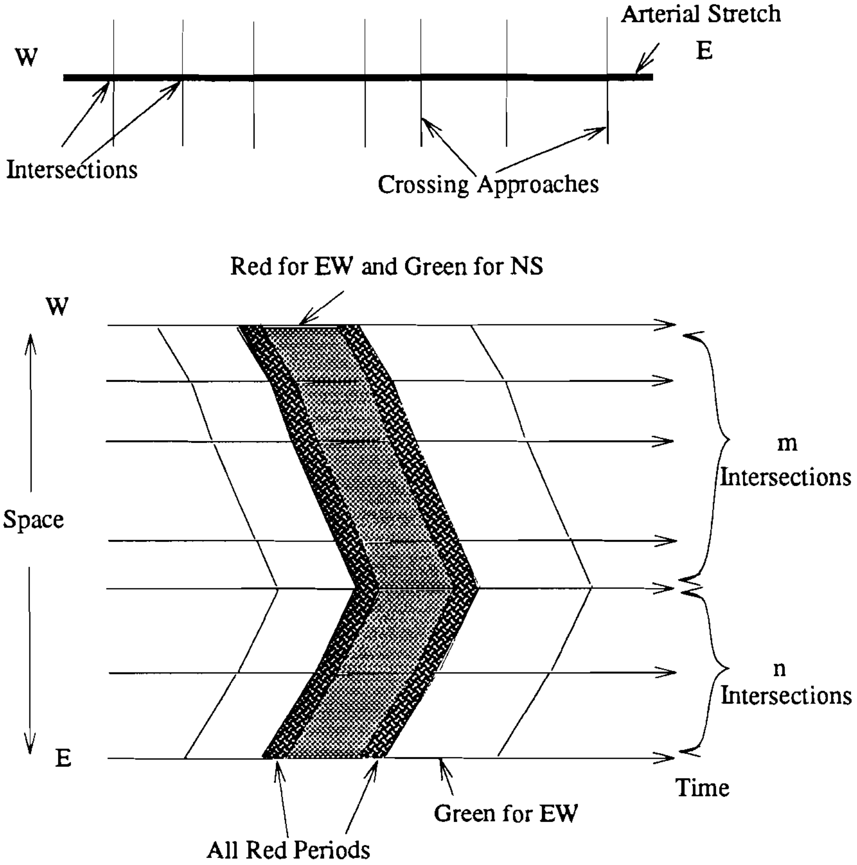

Figure 6: Example of a m:n progression

that area and preventing the congestion from spreading any further.

### Regional Controller Algorithm

The regional controller periodically checks the flow pattern changes in the traffic network and detects any traffic path which shows a rapid increase in flow. The detected path is then checked to see if it is safe to provide a progression and if so, a progression factor, which indicates the offset timings, is provided to the signal controllers in that path. If the above detected path shows signs of creating congestion then a congestion control factor, which is a restriction of the green length for the traffic flowing along that path, is provided to the signal controllers.

The regional controller performs the following steps to provide the factors required for controlling congestion and providing progression.

The regional controller sends a message to the signal controllers in its region requesting the load information of all their approaches periodically. The regional controller on receiving the data from the signal controllers, forms a Traffic Information Graph (TIG), in which each node represents a controller of an intersection and a directed link between two nodes represents the approach between those intersections. The links in the TIG are weighted by the traffic load level which is sent by the signal controllers. The TIG is now used to identify the critical and outstanding paths in the traffic network. A critical (outstanding) path is a sequence of links which are linked in the graph and which have a critical (outstanding) load level. All such paths are identified by tracing the links in the TIG which indicate a critical or outstanding load level. Once the critical and outstanding paths are detected the control factors are computed for each path, *i.e.* if it is a critical path then a restriction of the green lengths along that path is enforced and if it is an outstanding path a progression factor which indicates the offset timings is computed for the signal controllers in that path. The computed progression and control factors are sent to the signal controllers in that path.

### Exceptional Conditions

During exceptional conditions like occurrence of accidents, variable traffic flows, and failure of signal controllers, the performance of the traffic control system may be reduced. Hence, these exceptional conditions have to be detected and dealt with appropriately to provide better control.

The consequence of accidents is usually a reduction of flow rate in the affected approach, such that the flow rate downstream to the accident is low compared to the flow upstream of the accident. At the signal controller, the incoming and outgoing flows can be measured from the entry and exit detectors respectively and the difference in flows can be detected. If the flow rate drops rapidly at the exit detector but the entry detector does not indicate any decrease, then we can defensively assume that there has been a blockage in the approach. This is then reported to the regional controller during the periodic data collection phase of the regional controller. The regional controller then provides appropriate control factors to the signal controllers in and around the accident zone, to prevent congestion.

The detection of variable traffic is done by comparing the immediate past traffic levels of the approach with the current traffic level and a large difference may imply variable traffic. These variable traffic flows which might impede any outstanding paths are identified by the regional controller and a priority for progression is given to the outstanding paths, thus providing them with uninterrupted flow. In some cases two outstanding paths may cross each other hence the intersection at the crossing point may get a heavy influx of traffic. In this case the regional controller computes an optimal split and offset for the signal controllers for both these outstanding paths to make the flow in both paths smoother. This ensures that the traffic in the both these outstanding paths will flow without causing any congestion but however a small delay will be experienced by the vehicles in that path.

The regional controller also maintains preset timing plans for each signal controller and these plans are updated regularly. These preset timing plans reflect the immediate past traffic status and hence will perform effectively to current traffic. The failure of a signal controller will initiate a preset timing plan which is provided by the regional controller depending upon the time of day. The neighboring signal controllers of the failed controller are informed of the failure and these controllers restrict their flow into the affected intersection. Since the failed controller will not be able to provide its neighboring controllers with the flow rate and load information needed for their signal split computation, the regional controller provides this data which is estimated by using the history traffic flow data of the failed controller.

## 4 SIMULATION RESULTS

A simulation of the ARTC was performed and compared with the results of an optimized fixed time signal control plan. The timing computation for this fixed time control was done off-line with the use of the TRANSYT7F program to provide the cycle and split durations for specified arrival rates and geometry of the intersection. TRANSYT is an offline optimization program to produce signal timing plans for a traffic network [TRANSYT].

The model which we simulated consisted of an arterial with four intersections, as shown in Figure 7. Three lanes were provided for each approach, with the left lane being exclusively used for vehicles making left turns and the right lane being shared by vehicles turning right and straight ahead. The center lane was used by vehicles moving straight ahead only. All vehicles were modeled to travel at a uniform fixed speed and were allowed to change lanes. However, one segment of the arterial stretch, indicated in Figure 7, was designed

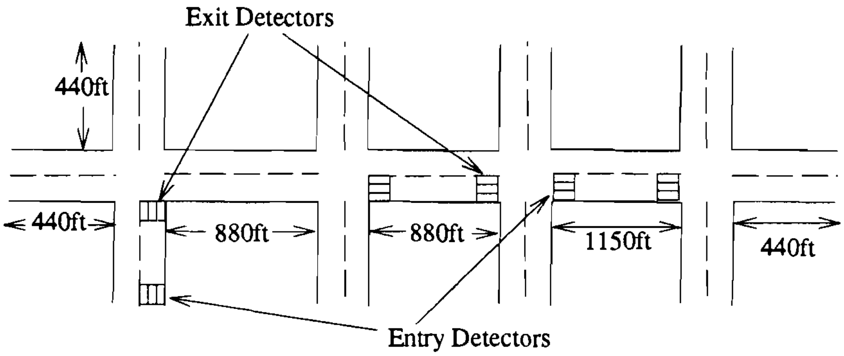

Figure 7: Geometry of Traffic Network Simulated

with a length higher than the rest of the links in the arterial stretch thus making the model asymmetric. Entry detectors were placed at the entry point of each lane and exit detectors were placed at the exit point of each lane, which are shown in Figure 7. The arrival rates of the vehicles, which is specified as an input parameter, is generated at the prespecified fixed rate. For example, if it was specified that an average of 400 cars per hour would enter a lane from the exterior of the system, then on a random basis 400 cars would be generated per hour

A simple two-phase sequence was used in the timing plan, and two all red periods, which were 3 seconds long, were provided between each phase. This 3 seconds of all red represented the loss time experienced by the vehicles during each phase change.

### Online Control Strategy

The online control strategies simulated were the two methods explained in the previous section, where one method used the queue estimation based on the CFP, which we called the CFP based, and the other used the flow rates for the queue estimation, which we called the flow based. The simulation program for the CFP based approach, estimated the queues in the arterial as explained in the previous section and for estimating the queues in the crossing approaches it used the flow based. The signal timing plans were computed at each controller in such a way that a progressive service would be provided for the traffic flow which is heavy. In our model the arterial stretch carried a heavier flow of traffic most of the time and hence was provided with a progressive platoon service.

Moreover, the online control method used by ARTC performed the following steps to control congestion, in addition to the steps described in the previous section to compute the split. The following steps are executed when any direction is to be provided with a green phase.

1. Obtain information from neighboring controllers about the number of vehicles present in the link between these intersections.
2. If the neighbor's load is lower than a $THRESHOLD$ level, which is a value of the allowable load, then it is assumed safe to provide a progression and the split is computed and executed.
3. If the neighbor's load exceeds the $THRESHOLD$ value, then a congestion control is enforced by bounding the split time such that the number of vehicles which can pass in the split is less than or equal to the $ALLOWANCE$ value. This $ALLOWANCE$ value is fixed by the neighbor who reports a low exit rate.

The performance of the traffic control methods described above are compared based on the following performance indices.

- **Average Delay per Vehicle:** A vehicle comes to a stop at a intersection if the signal light for its direction is red and hence waits at the intersection until it changes green. This waiting time at the intersection is the delay experienced by the vehicle and should be minimized. The average delay per vehicle is the total delays of the vehicles in the traffic network divided by the total number of vehicles which had been generated in the system. This value of the average delay is the waiting time which a vehicle will experience in the traffic network.

$$\textit{average-delay} = \frac{\sum \textit{delay-for-each-vehicle}}{\textit{total-number-of-vehicles}}$$

- **Mean number of stops:** This parameter can account for the average number stops a vehicle has to experience while traveling through the road network which is under control by the traffic control system being simulated. The formula to compute the mean number of stops is given below, where $n$ is the total number of vehicles.

$$\textit{mean-number-of-stops} = \frac{\sum \textit{number-of-stops-for-each-vehicle}}{\textit{total-number-of-vehicles}}$$

The simulation for both the systems was performed with different sets of traffic flow data for the approaches. The two main cases under which the system was simulated were, a fixed traffic flow rate and a varying traffic flow rate. The fixed traffic flow rate, generated vehicles on all approaches at a predefined rate on a random basis for intervehicle gaps. The varying traffic flow rate, generated vehicles at all approaches such that the arterial stretch varied from 30% of a predefined flow rate value to 170% and the crossing approaches were varied from 5% of the predefined rate to 50%. The percentage of vehicles turning left from the arterial to the crossing approaches was set at 10% and 2% for each of the fixed and varying flow rate simulations, thus providing us with a total of four different cases. For each of these cases, the TRANSYT program was run to produce the signal timing plan, by specifying the arrival rates and probabilities at which the vehicles were expected to turn.

The performance result graphs of these parameters plotted against time are shown below. As can be seen from the plots of delay against time, the performance of ARTC is significantly better than TRANSYT with a lower delay value in the varying arrival rate. As the percentages of the crossing approach traffic increases, ARTC performs better with the CFP based and the flow based methods. This varying traffic can be observed in reality, where the traffic flow in the network fluctuates depending on time of day and other factors, and we believe that the ARTC system would perform much better and can exploit its use of on-line split computation in these situations. With a fixed arrival rate, we can see that the flow based ARTC and TRANSYT perform much better than the CFP based ARTC, which is because the CFP based ARTC uses the delay time as the performance index and hences provides shorter cycle times. Moreover, for a fixed arrival rate, a flow based approach will provide the necessary split which will be sufficient for this average fixed flow. Moreover, simulation results indicate that as the traffic flow increases, in the crossing approaches for the fixed flow cases, the performance of both ARTC methods and TRANSYT produce almost similar values of delay and stops.

It can also be seen that the number of stops increases with increasing flow rate values. This is due to the fact that the vehicles approaching the intersection in the East-West direction are provided a progression when the North-South flow rate is low, but as it is increased the East-West direction is not always provided a progression. Moreover, a shorter cycle length would also increase the number of stops and is the reason why the number of stops for TRANSYT increases as the North-South flow rate increases. This improved performance of ARTC indicates that an on-line control is better for traffic control with fluctuations in the arrival patterns and a flow based ARTC is better for a fixed arrival pattern.

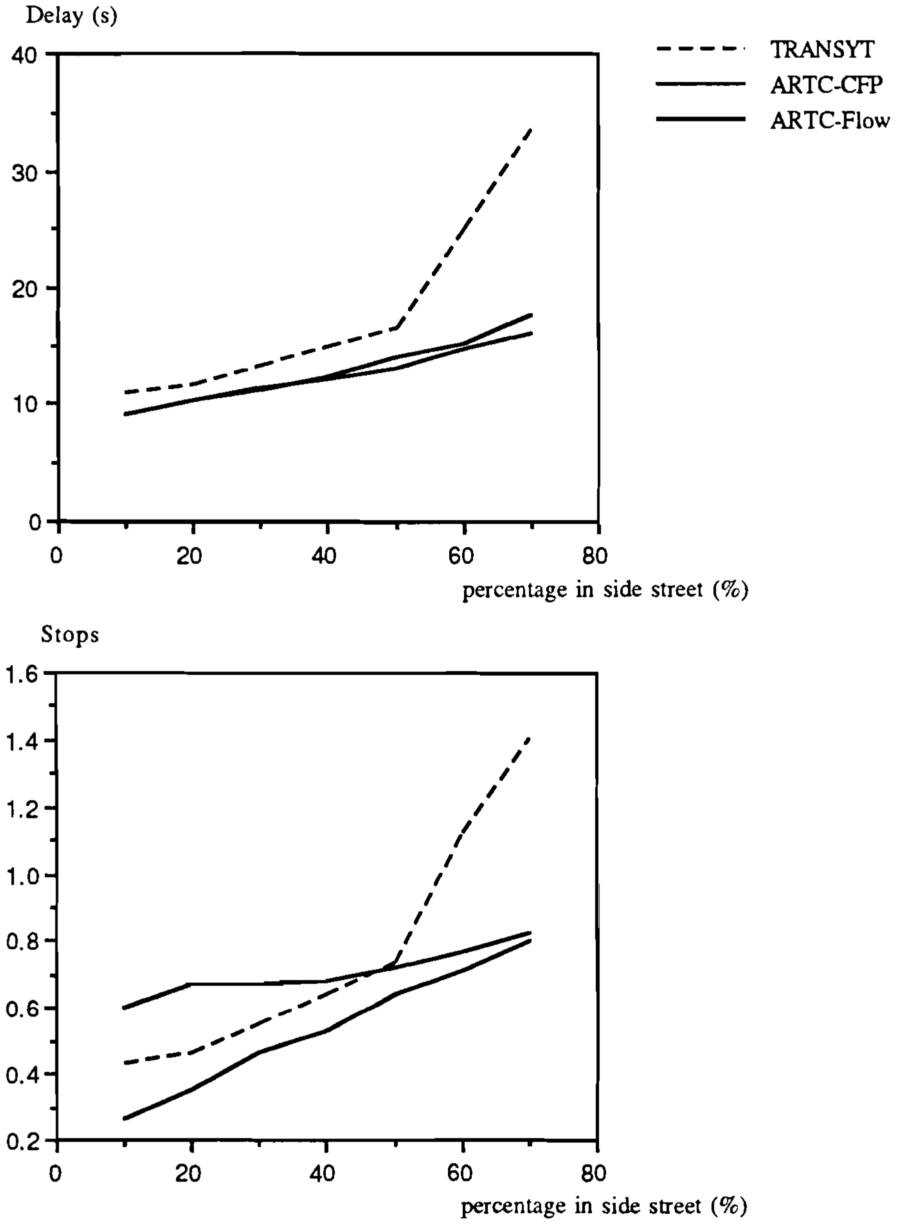

Figure 8: Varying Arrival Rate With 2% Left Turn From Arterial

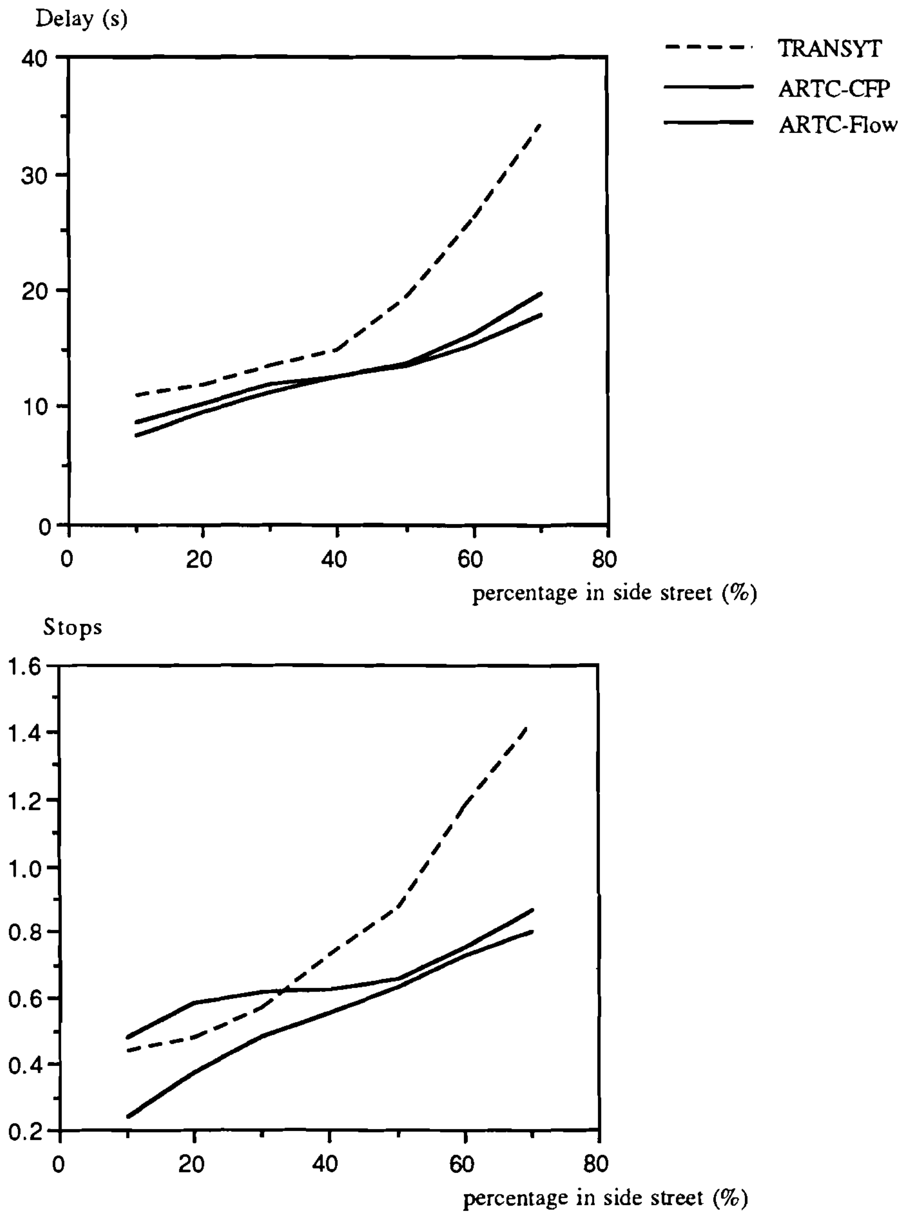

Figure 9: Varying Arrival Rate With 10% Left Turn From Arterial

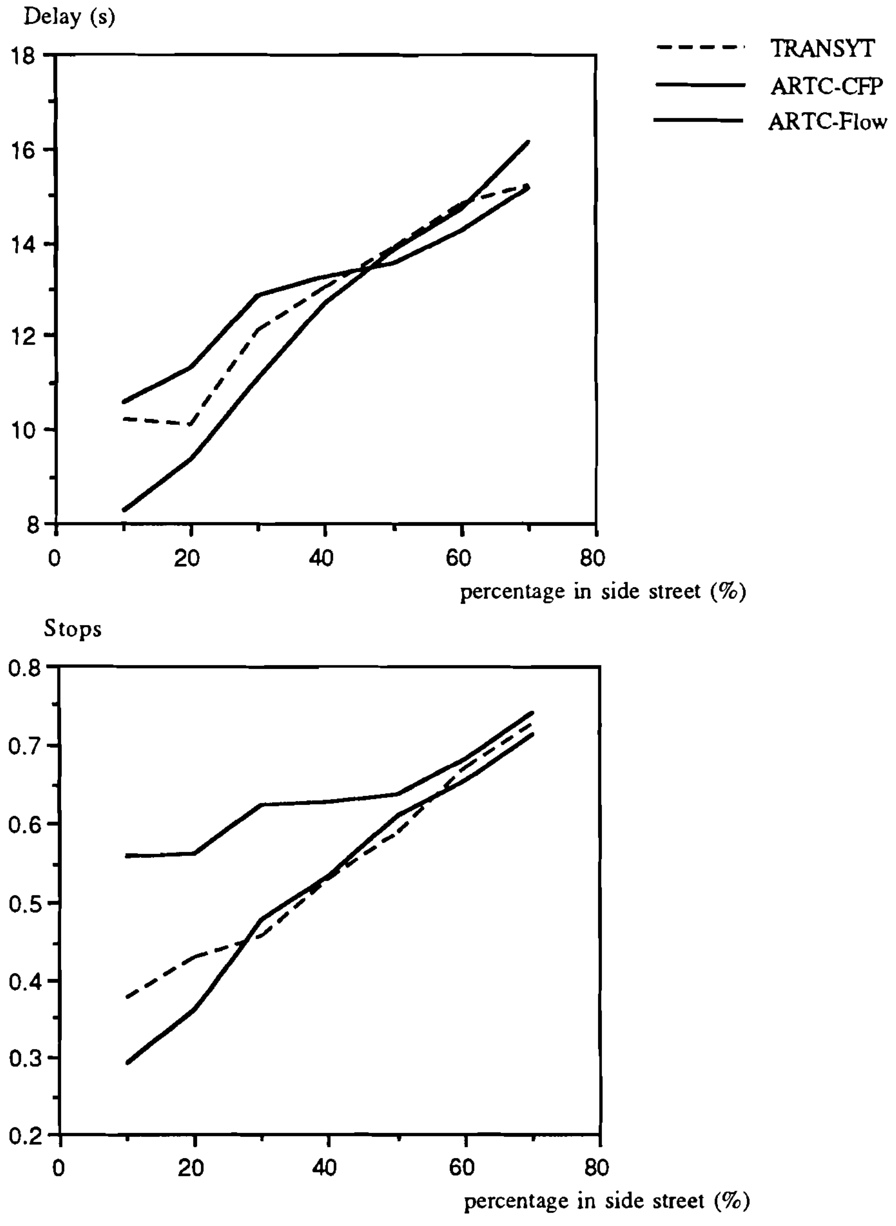

Figure 10: Fixed Arrival Rate With 2% Left Turn From Arterial

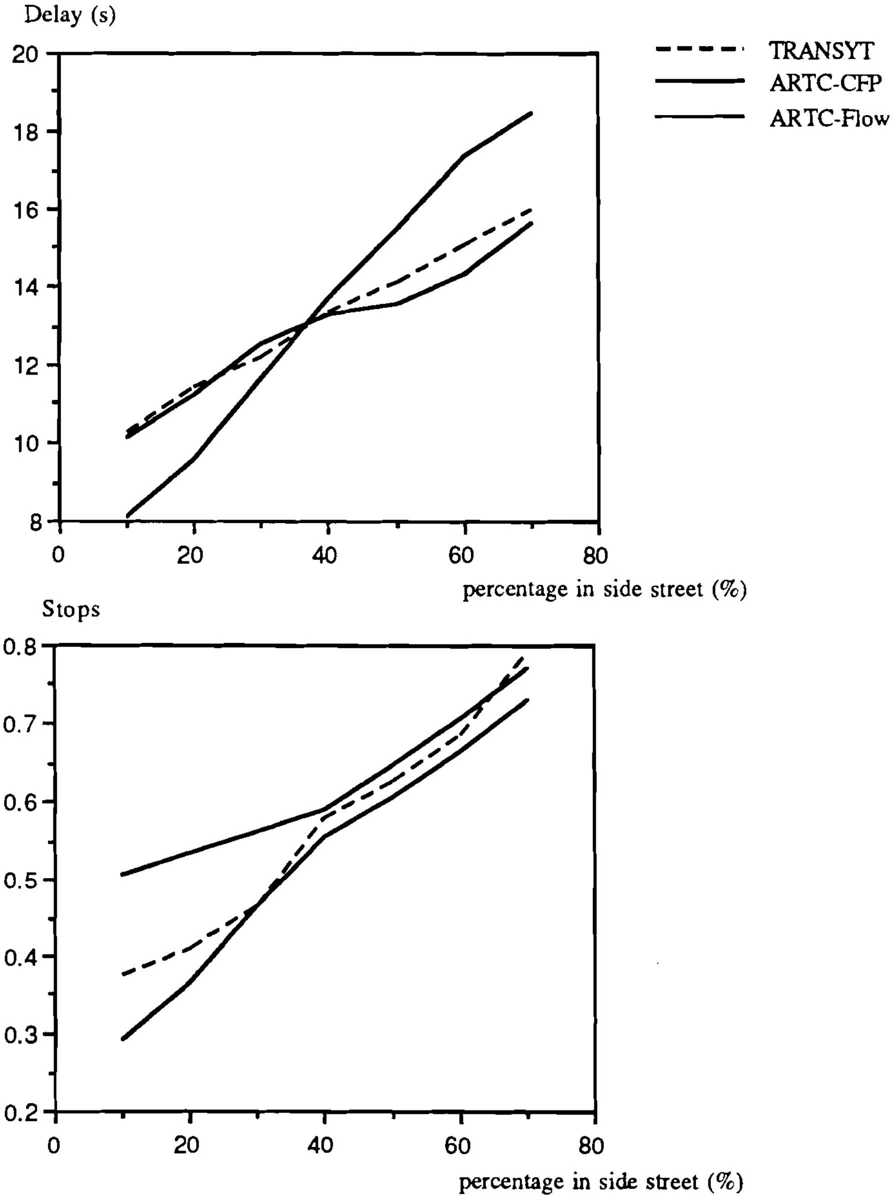

Figure 11: Fixed Arrival Rate With 10% Left Turn From Arterial

# 5 CONTROLLER AND NETWORK DESIGN

Signal controllers are expected to perform computations correctly even during acute weather conditions, thus it is necessary to use hardened microelectronic components in the signal controllers which have a better chance of survival in these conditions than their regular counterparts. The main computational outputs of the signal controllers are to set the signal timings in terms of the physical time. Moreover, the CFP data must be frequently updated within a physical time window, hence, the microprocessors used for the signal controllers should have an accurate measure of the physical time. In conventional signal controller design, the physical time is measured by some hardware timers, external to the microprocessors, which are used to indicate the physical times of events like time-out, beginning of the green phase, a new computation iteration, etc. The number of chips will be fairly high with the use of regular microprocessors and discrete logic circuits, and this may lead to a high component failure rate in the ARTC system. On the other hand, the current state-of-the-art 32-bit microcontrollers have over 15 different timers/counters built-into the same chip and in addition, have on-chip memory, and have low power operation mode, etc. Failures like power loss, transient noise, etc., are detected by these mechanisms and after these disruptions cease, the application programs can be recovered without special software routines. Since we do not need any external timers/counters and their control logic, the chip count, and the component failure rate is reduced. Thus, to have efficient time management with a low chip count, the microcontrollers are used in ARTC signal controllers.

## 5.1 Signal Controller

Conceptually, the signal controllers are designed in a modular fashion consisting of four modules called the input module, which reads the detector data, the computation module, which computes the signal timing plans, the output module, which controls the signal lights, and the communication module which provides inter-node communication. Each module consists of a microcontroller which is used as the master of the module. To achieve maximum system reliability in the signal controllers at a reasonable cost, a hybrid redundant system architecture, in which one of the modules is replicated and less expensive fault-tolerance designs applied to other modules, is used. In ARTC, the computation modules are replicated to provide on-line backup and serve as the system hard core to correctly detect failure of cheaper fault-tolerance designs.

The design of signal controllers emphasizes on prolonging the lifetime of the signal controllers and controlling the behavior of a totally or partially failed signal controller. The system cost will increase substantially if we attempt to further increase the degree of fault coverage in signal controllers. Thus, a network level fault-tolerance strategy is designed to prevent failed signal controllers from providing inconsistent information to their neighbors. In the network level fault-tolerance technique, signal controllers test each other for faults periodically. The signal controllers report all testing outcomes to the regional controller. This confirmation is then broadcast to every node in the region thus preventing any malicious message from the failed node to tamper the processes on other active nodes.

## 5.2 ARTC Network

Although currently optical fiber is more expensive than other communication media, it has the highest reliability and extensibility in the long run [18]. Thus, the optical fiber is used as the communication medium for ARTC networks.

In ARTC, the signal controllers exchange current traffic flow information in real-time, hence the communication delay should not cause excessive distortion to this information.

The communication delay is mainly determined by the message processing time in the communication modules of signal controllers. For example, a vehicle entering a 600ft long street and moving at $50mph$, will travel 15 feet in 0.2 seconds and for the information distortion to be less than 3%, the message transmitted to an adjacent controller should be delivered before 0.2 seconds. If we assume that the message is routed through five signal controllers then the message processing time at each controller should be less than $\frac{0.2}{5} = 0.04$ seconds, but in reality it takes 0.005 seconds to process one message at a controller and hence a transmission speed of approximately 0.003 seconds has to be achieved between two directly connected controllers. So for a 10 byte message, *i.e.* 80 bits long, the transmission speed required is 26k bits per second. The lower bound for the transmission speed, which is calculated solely based on data exchanges, should satisfy the real-time communication constraints even when some nodes or links in the network fail. But during these failures the messages must be rerouted in the network leading to an additional delay, hence the transmission speed for the network should be sufficiently high to satisfy the real-time requirements.

An important issue in the message rerouting is that the messages may be delivered out of order in the destinations and hence may lead to an inconsistent state in the system [19]. To avoid these misordering of messages, we use the timestamp method to establish a global consistency and to discard aged messages when necessary. The number of discarded messages due to expired deadlines is minimized when the message with the earliest expiration time is given the highest transmission priority.

The message routing is a major computation burden to the communication modules of the signal controllers. If the system does not have a systematic addressing scheme then the message routing direction must be determined from a *message routing table* in the signal controllers. It is very expensive and difficult to maintain a large number of message routing tables in ARTC signal controllers. With the systematic addressing scheme, the message routing direction can be calculated by using the addressing scheme without using any message routing table. The systematic addressing scheme is also important to the expansion of the network and is hence used by ARTC, as explained below.

**Addressing Scheme and Fault-Tolerant Routing:**

Most urban road networks resemble a regular mesh with a few irregular branches. In designing the network we utilize this natural property of the existing road structures and decompose ARTC's network into two parts: the mesh network and the irregular branches. For a regular mesh structure road system a full mesh network could be directly used for the inter-node communication between signal controllers, but the cost of a full mesh network is very high and the utilization is very low. The other option is to connect all nodes with a linear cable, but the resulting network is unreliable and is difficult to meet the real-time communication requirements. Hence, to reduce the cost, some links in the full mesh network have to be removed but the resulting network should preserve the real-time communication and fault-tolerance requirements. The ARTC network is hence designed by initially adding vertical links to all the controllers in the network, and then horizontal links are added one by one until the communication delay is reduced to a level that can satisfy the real-time requirements. Each point of connection in the network with a signal controller is called a *node*. The irregular branches, to be added in the network, are divided into groups called *branch groups.* At least one node in a branch group is connected to a node $N_m$ on the mesh, where $N_m$ is called the parent of the branch group. This resulting network is called a *sparse mesh network* (SMN), and this is shown in Figure 2.

The SMN consists of *cross nodes* and *chain nodes* where the cross nodes are directly connected to four of the nodes and the chain nodes are directly connected to two nodes which lie in the sparse mesh structure. Some signal controllers which are adjacent to each other and lie on two adjacent vertical links might be chain nodes, which means that they are not directly connected. Hence these controllers exchange information by passing messages along some intermediate signal controllers.

A two-layer hierarchical addressing scheme is used for nodes in the mesh and the branch groups. The first layer of addressing is reserved for nodes in the mesh, and the second layer of addressing space under a mesh node $N_m$ is reserved for the branch group connected to $N_m$. All messages that must leave or enter a branch group has to pass through its parent node, and each branch node uses a table for message routing.

On the SMN, not all adjacent nodes have direct links, and thus message routing is divided into the micro and macro levels. At the micro level routing, a message received by a chain node $N_c$ is first checked to see if the destination of the message is $N_c$, if it is not then the message is sent to the neighbor which did not send that message to $N_c$. The macro level routing is performed by the cross nodes to find the direction of the next cross node for each message using the full mesh addressing scheme. The full mesh addressing scheme uses the relative positions of the source and destination in terms of the x-coordinate and y-coordinate and the optimal routing direction is determined.

Blocked messages should be rerouted, when a node or a link fails. If a message is sent by a cross node to a direction where a node or a link has failed then the condition is detected by the node immediately adjacent to the failed node or link. The condition of the failure is piggybacked on the message and is returned to the cross node from which the message originated. The message is then rerouted by the cross node and all other messages which arrive at the cross node and have to pass through the blocked connection are rerouted by the cross node until the blockage is cleared. Moreover, a time-out parameter $T$, which is the maximum number of hops that a message can take, is added and messages are discarded if the number of hops the message has taken exceeds $T$. This fault-tolerant technique is simple, effective and guarantees delivery of messages as long as a path exists between the source and destination nodes.

# 6 CONCLUSION AND FUTURE DIRECTION

In this paper we presented the ARTC system which provides effective congestion prevention and areawide progression. One of the major advantages of employing the ARTC is that it prevents congestion rather than detecting it after it has occurred. Moreover, the progression of traffic can be achieved at a high level in the ARTC, since the signal controllers are able to communicate with each other. Even though the above aspects of traffic control can be achieved to a sufficient level by the communication between signal controllers, the regional controller is provided for efficient areawide traffic control coordination. Since the regional controller has a global view of the traffic network, it provides non-linear progression and also detects anomalies such as accidents and traffic fluctuations. Two methods to compute the cycle and split were suggested and the use of $CFP$ and traffic flow show that we are progressing in both, the vehicle location based traffic control and the flow based approach. From the simulation results it can be seen that ARTC performs better than the fixed time control plan which was computed by TRANSYT.

The failure of a signal controller does not affect the performance of the traffic control system in a drastic way, since the regional controller provides the necessary data for signal timing computations. But to provide better fault-tolerance, the signal controllers are designed in a modular structure with the computation module being replicated. For the communication between signal controllers, the sparse mesh network is designed at a reasonable cost such that it satisfies the fault-tolerant and real-time communication requirements.

With minor modifications to the controllers it is possible to expand the ARTC system to accommodate the functions to be provided by the Intelligent Vehicle-Highway System (IVHS). The computation methods for the control algorithms in the ARTC system is still under refinement to improve the performance. The traffic network which was simulated was small and hence did not reflect the full capability of the ARTC control. Currently the simulation is being performed for a wide traffic network and the use of the regional controller, which is currently not used, can then be exploited.

## ACKNOWLEDGEMENT

The study and development of the ARTC system has been made possible by many people. Among them we particularly thank Mr. Gary Tretsch, who is with the Texas State Department of Highways and Public Transportation, and Mr. Sadler Bridges, who is with the Texas Transportation Institute, for their financial and technical support for this study. We also acknowledge the constructive comments made by Ms. Taesoon Park and Mr. Ying Hao.

# References

[1] D. A. Haver and P. J. Tarnoff, "Future Directions for Traffic Management Systems," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 4–10, February. 1991.

[2] F. B. Lin, "Operational analysis of traffic actuated signal control," *Traffic Engineering and Control, Vol 24*, no. 12, pp. 573–579, 1983.

[3] W. S. Homburger and J. H. Kell, *Fundamentals of Traffic Engineering*, Institute of Transportation Studies, University of California, 1984.

[4] A. Skabardonis and A. D. May, "Comparative analysis of computer models for arterial signal timing," *Transportation Research Record 1021*, pp. 45–52, 1985.

[5] S. L. Cohen and J. R. Mekemson, "Optimization of left-turn phase sequence on signalized arterials," *Transportation Research Record 1021*, pp. 53–58, 1985.

[6] N. M. Rouphail, "Analysis of transyt platoon-dispersion algorithm," *Transportation Research Record 905*, pp. 72–83, 1983.

[7] J. C. Williams, H. S. Mahmassani, and R. Herman, "Urban traffic network flow models," *Transportation Research Record 1112*, pp. 78–88, 1987.

[8] M. Sakita, "Timing design of traffic signals," *Transportation Research Record 1069*, pp. 83–87, 1986.

[9] S. L. Cohen and C. C. Liu, "The Bandwidth-Constrained transyt signal-optimization program," *Transportation Research Record 1057*, pp. 1–9, 1986.

[10] D. I. Robertson, "Research on the TRANSYT and SCOOT Methods of Signal Coordination," *ITE journal*, pp. 36–40, January 1986.

[11] D. I. Robertson and R. D. Bretherton, "Optimizing Networks of Traffic Signals in Real Time - The SCOOT Method," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 11–15, February. 1991.

[12] P. B. Hunt, D. I. Robertson, and R. D. Bretherton, "The SCOOT on-line traffic signal optimization technique," *International Conference: Road Traffic Signalling*, no. 207, pp. 59–62, 30 March - 1 April. 1982.

[13] J. Y. K. Luk, "Two traffic-responsive Area Traffic Control methods: SCAT and SCOOT," *Traffic Engineering and Control*, vol. 25, no. 1, .

[14] P. R. Lowrie, "The Sydney co-ordinated adaptive traffic system - principles, methodology, algorithms," *International Conference: Road Traffic Signalling*, no. 207, pp. 67–70, 30 March - 1 April. 1982.

[15] N. H. Gartner, "OPAC: A Demand-Responsive Strategy for Traffic Signal Control," *Transportation Research Record*, no. 906, pp. 75–81.

[16] E. Rowe, "The Los Angeles Automated Traffic Surveillance and Control (ATSAC) System," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 16–20, February. 1991.

[17] D. I. Robertson, "Cyclic flow profiles," *Traffic Engineering and Control*, pp. 640–641, 1974.

[18] G. R. Elion and H. A. Elion, *Fiber Optics in Communications Systems*, Marcel Dekker, Inc., New York and Basel, 1987.

[19] M. Schwartz, *Telecommunication Networks, protocol, modelling and analysis*, Addison-Wesley Publishing Company, 1987.