TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
TX-90/1245-4		

4. Title and Subtitle	5. Report Date
Area Wide Real Time Traffic Control--Intermediate Report	February 1992
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Junguk L. Kim, Steve J-C. Liu, Ying Hao, Taesoon Park, Prabaran Swarnam	Research Report 1245-4

9. Performing Organization Name and Address	10. Work Unit No.
Texas Transportation Institute Texas A&M University System College Station, Texas 77843	11. Contract or Grant No. Study No. 2-1-90/1-1245

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Texas Department of Transportation P.O. Box 5051 Austin, Texas 78701	Interim September 1989-August 1990
	14. Sponsoring Agency Code

15. Supplementary Notes

Research performed with all State Funds.
Research Study Title: Txas Advanced Transportation Technology Project

16. Abstract

This report is an introduction to the type of traffic signal control system that may be used in urban areas where "real-time" control, of traffic adaptive, system is justified. The system described in the report is a step beyond current real time systems such as SCOOT and SCAT. The traffic control strategy described is known as ARTC or Area-wide Real-time Traffic Control system and is an advanced design of network traffic control which uses "smart" controllers as each intersection and a regional of centralized control system. The system is designed to be both responsive to the traffic at each individual intersection from feeding congestion levels of traffic to those parts of the network which do not have the capacity to handle the traffic. The system is described in this report.

17. Key Words	18. Distribution Statement
Traffic Signal, Traffic Adaptive, Real-time Signal Control, Traffic Management Center	No Restrictions. This document is available to the public through the National Technical Information Service 5285 Port Royal Road Springfield, Virginia 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	53	

Form DOT F 1700.7 (8-69)

THE AREA-WIDE REAL-TIME TRAFFIC CONTROL (ARTC): THE CONCEPTS

by

Junguk L. Kim
Assistant Research Specialist

Steve J-C. Liu
Assistant Research Specialist

and
the following Research Assistants

Ying Hao

Taesoon Park

Prabaharan Swarnam

Texas Transportation Institute

Research Report 1245-4

Research Study Number 2-1-90/1-1245
Texas Advanced Transportation Technology Project

Sponsored by

Texas Department of Transportation

Texas Transportation Institute
The Texas A&M University System
College Station, Texas 77843

February 1992

Abstract

This report presents a traffic control system, called the Area-Wide Real-Time Traffic Control (ARTC) System, which addresses occurrences of congestion and provides areawide traffic progression. The signal controllers in ARTC are interconnected through a communication network and by exchanging traffic flow information among them, ARTC provides a new concept in areawide traffic control. The traffic area controlled by ARTC is divided into regions and a regional controller is provided for each region. The regional controller periodically collects traffic flow information from the signal controllers to view the traffic flow of its region and anticipate congestion by detecting changes in traffic flows. Once congestion is anticipated in a region, the influx of traffic into this region is reduced until the congestion situation disappears.

The signal controllers and the communication network are designed to support the real-time traffic control. Fault-tolerant designs for the signal controller and the network are also presented.

Disclaimer

The contents of this report reflect the view of the authors who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Texas Department of Transportation or the Federal Highway Administration. This report does not constitute a standard, specification, or regulation. In addition, this report is not intended for construction, bidding, or permit purposes. This report was prepared by Junguk L. Kim, Steve J-C. Liu, Ying Hao, Taesoon Park and Prabaharan Swarnam.

Summary

In this report, we characterize the issues involved in developing an efficient areawide traffic control system. The objective of our study is to develop a system which provides orderly movement of traffic, shortens average delay to vehicles, reduces congestion occurrences, and improves the utilization of the roads. In our system under development, called the Area-wide Real-time Traffic Control system (ARTC), all computing nodes, called signal controllers, are connected to each other. The ARTC is essentially a very large scale distributed computing system. The signal controller uses data from the vehicle detectors placed in all approaches to the intersection under its control. Each controller communicates through the computer network with its neighbors for better utilization of the roads. Failures of regional/signal controllers or the computer network may immediately degrade the service quality of the traffic system. To maintain continued service in the face of failures, signal controllers are designed with a modular organization and redundant hardware.

In the ARTC system, traffic overflow into an approach is prevented by exchanging information between signal controllers incident on the approach on current capacity of the approach. Such information is collected from detectors installed in all approaches of signal controllers. This information exchange between signal controllers provides the real-time signal timing plan for progressive traffic. However, the data collected from detectors cannot always be perfect, hence total reliance on the collected information for congestion control may be dangerous. For this reason, a group of the signal controllers are connected to a powerful computing node, called the regional controller. Each traffic path carrying a considerable traffic flow, such as an arterial street, is detected by the regional controller. If this progressive flow reaches a critical level which may cause congestion, then the regional controller orders the signal controllers in that path to reduce the flow. With these mechanisms, we believe that the throughput of the traffic network can be significantly improved. It is also possible to utilize this regional controller to provide a better signal timing plan for a progressive traffic.

Contents

List of Figures

1 INTRODUCTION

Due to the progress of economic growth, there has been an enormous increase of automobiles on the roads in recent years. This increase in the number of vehicles on the roads has created alarming levels of congestion. As a result, millions of person-hours have been wasted. Moreover, we face many problems such as increased air pollution, travel delay, and waste of fuel.

The traffic congestion may occur due to certain traffic conditions, such as peak-hour traffic, accidents, road-side friction, bottlenecks, and surges in traffic flow. During peak hours, the number of the vehicles entering a road from side streets, parking lots, etc. is much higher than that of the ones which leave the road; hence the road gets filled resulting in congestion which may spread rapidly throughout the network of roads. If an accident occurs in a lane, the traffic flow of the road reduces, resulting in the flow of the neighboring roads being affected undesirably. The flow can be also reduced due to road-side parking, bus stops, road construction, loading activities etc. This indicates that the capacity of a road changes dynamically depending upon the state of the road. The flow of traffic may also be hindered due to a road of lesser capacity in the path of the flow. This road limits the rate of flow for the entire path. When there is an unexpected surge of heavy traffic onto a road it changes the existing flow pattern considerably. If the traffic control system does not respond appropriately in 'normalizing' this change of pattern, it might result in congestion.

Current traffic control systems have not been able to utilize the road capacities efficiently. To the best of our knowledge, the successful traffic systems installed can optimize traffic flow to a limited degree, such as signal coordination for the progression in an arterial street [1, 2, 3]. One of the common problems of the current traffic control systems is the

ineffective adaptation to transient traffic volume change. When surges in traffic flow or abnormal traffic overflow occurs, it is quite likely that the roads are congested under the current traffic control systems, since the system cannot adapt quickly and efficiently to such sudden traffic changes. In the worst case the traffic might come to an indefinite stand-still in some roads, called deadlock. One important reason for these problems is that a major number of the current traffic control systems do not have an effective areawide coverage of a traffic network consisting of freeways, arterials and cross streets.

The traffic control systems presently employed fall into two categories: *preset (fixed-time) control* and *traffic-actuated control.* The fixed-time control systems rely on historical data to prepare timing plans for a signalized area. Timing plans are stored in memory which are installed in each controller. A particular plan is switched into operation according to time of day. A computer program is employed to optimize timing plans for each intersection off-line. Vehicle detectors are not required in this method. The major factors taken into consideration when designing a fixed time plan are statistical data for the peak-hour traffic and road side friction. Hence it cannot respond to the dynamic changes efficiently.

The traffic-actuated control systems calculate signal control parameters according to prevailing traffic conditions. This method requires information about the current flow of traffic and even about the impending flow of vehicles to an intersection. Hence it requires vehicle detectors on all approaches. There are many forms of implementation of the traffic-actuated system and among these SCAT [1] and SCOOT [2] are noteworthy. The common factor which is taken into consideration by most of these traffic-actuated systems is the traffic volume (which includes peak-hour traffic, road side friction, slow moving vehicles and surges in traffic flow). One of the deficiencies in these systems is that they cannot efficiently restrict the flow into congested areas due to lack of areawide real-time traffic

status information. We discuss other limitations of these systems in Section 2.

In this report, we characterize the issues involved in developing an efficient areawide traffic control system. The objective of our study is to develop a system which provides orderly movement of traffic, shortens average delay to vehicles, reduces congestion occurrences, and improves the utilization of the roads. In our system under development, called the Area-wide Real-time Traffic Control system (ARTC), all computing nodes, called signal controllers, are connected to each other. The ARTC is essentially a very large scale distributed computing system. The signal controller uses data from the vehicle detectors placed in all approaches to the intersection under its control. Each controller communicates through the computer network with its neighbors for better utilization of the roads. Failures of regional/signal controllers or the computer network may immediately degrade the service quality of the traffic system. To maintain continued service in the face of failures, signal controllers are designed with a modular organization and redundant hardware.

In the ARTC system, traffic overflow into an approach is prevented by exchanging information between signal controllers incident on the approach on current capacity of the approach. Such information is collected from detectors installed in all approaches of signal controllers. This information exchange between signal controllers provides the real-time signal timing plan for progressive traffic. However, the data collected from detectors cannot always be perfect, hence total reliance on the collected information for congestion control may be dangerous. For this reason, a group of the signal controllers are connected to a powerful computing node, called the regional controller. Each traffic path carrying a considerable traffic flow, such as an arterial street, is detected by the regional controller. If this progressive flow reaches a critical level which may cause congestion, then the regional controller orders the signal controllers in that path to reduce the flow. With these mecha-

nisms, we believe that the throughput of the traffic network can be significantly improved. It is also possible to utilize this regional controller to provide a better signal timing plan for a progressive traffic.

This report is organized as follows. In the next section, we discuss the signal control parameters to identify shortcomings of the current systems. In addition, we describe the logical structure of the ARTC system. The computational algorithms for signal and regional controls are presented in Section 3. We show the development of the algorithms in stages. First, we present the algorithm under an assumption that conditions of the ARTC system are ideal, *i.e.* all detectors and controllers are perfectly reliable, no accident occurs, and there are no parking lots in the area under the ARTC system. The algorithms is then evolved to handle more severe conditions of the road, such as presence of parking lots, accidents, etc. Section 4 discusses the structural design of the signal controller and the communication network among the controllers. We then summarize the results of our study and discuss future plans in Section 5.

2 BACKGROUND

In this section we discuss the current practices in a comparative manner to identify the benefits and shortcomings of these systems.

With the advent of computers, many sophisticated computer controlled signal systems were developed to cope with the increased traffic volumes. These systems have been trying to minimize the average delay time and the average number of stops for the vehicles. The overall traffic control has been largely dependent on each individual signal control. The following parameters have been used to characterize different signal control methods:

a) Cycle : The total time required for one complete sequence of signal phases.

b) Offset : The time difference between the start of the green indication at one intersection as related to the start of the green indication at the next intersection in a path of progressing traffic.

c) Split : The percentage of a cycle duration allocated to each of the various phases.

We can classify the current traffic control systems into two categories: the preset control (fixed time) system and the traffic-actuated control system. The preset controller assigns the right of way to each approach at an intersection according to a predetermined signal schedule, which has a fixed cycle length, split and change interval.

The preset controllers use off-line methods to compute the cycle, offset and split plans. The cycle length can be computed by using Webster's equation [4]. This is an iterative process over the equations which use the past traffic flow data compiled from traffic counts to reflect traffic volumes for the specified time of day. The plans are selected and executed according to time of day and day of week. These type of systems cannot respond appropriately and in time to unpredicted changes in traffic flow.

The traffic-actuated controllers make use of the data from vehicle detectors placed on the approaches to the signal and assign the right of way to each approach which varies with the changing traffic demand. The actuated controllers can be further divided into: (i) Semi-actuated controllers, (ii) Fully-actuated controllers, (iii) Volume-Density controllers, (iv) Lane-Occupancy controllers and (v) Adaptive controllers [5, 6, 7, 8, 9, 10, 11].

The vehicle detectors used for the semi-actuated controllers are usually placed in the streets that have a lesser traffic volume than the crossing street, so that the minor flow can actuate the signal. The green time on those streets with minor flow is designed to be fully

utilized while the rest of the cycle time is allocated to the main flow. Some implementations for signal control employ this method.

The detectors used for the fully-actuated controllers are placed in all the approaches to the intersection. Among those detectors, some detectors are used to provide a service in which the cars do not have to stop at the signal, if possible. These controllers have no fixed cycle lengths and splits and are used at intersections where the approaches have relatively equal volumes and the traffic volumes vary very often. If there is no traffic in a link, the signal does not serve that link. These types of control are usually used at an isolated intersection which satisfies the above volume requirements.

The volume-density controller tries to provide a green to the vehicles as long as the inter-vehicle gap between the vehicles passing the intersection lies under a maximum threshold. The cars waiting in the queue on the red phase form a group, called the platoon, when they are served by a green phase. The vehicles might be farther apart, due to varying speeds, while approaching the intersection which is in the red phase. The volume-density controller reduces this gap between the vehicles and makes the platoon more dense by changing the phase to green which depends on the count of the vehicle detections at the detectors in the approach with the red phase. The green phase is then extended depending upon the rate at which the vehicles are detected at the detectors in the approach with the green phase and the delay experienced by the vehicles at the red phase. This concept was used in some implementations of SCOOT and SCAT.

A lane-occupancy controller is often used for left-turn or other specific turning movement phases at actuated intersections, since those turning cars can impede the flow of the through traffic if they are not served in time. This control is designed to change the signal to green only if it senses the presence of a vehicle for a specified length of time or the presence of a

quite a large number of vehicles; the phase is terminated as soon as the vehicles leave the detection zone. Most of the actuated controllers use this concept.

The adaptive controller uses the detectors in a way similar to the fully-actuated controllers. The arrival patterns on all approaches are considered for signal setting. However, the length of a cycle is finite hence all the approaches will be served in a round robin fashion. The length of green and red phases of each cycle is calculated using the dynamic programming technique. An example system which uses the dynamic programming technique for an isolated intersection is OPAC [3]. When these signal controllers are interconnected with each other or through a central controller, information from the detectors is used to generate an optimal signal timing plan, hence they may provide better progression flow to the traffic than the fully-actuated controllers.

2.1 Most Successful Adaptive Control Systems

Among the systems implemented in various places, SCOOT and SCAT are known to have been successful [1, 2, 12, 13, 14, 15]. Here, we discuss the design of each of the three parameters as done by SCOOT and SCAT.

a. Cycle :

The traffic network under SCOOT is divided into regions and each region is controlled by a regional controller. In each region, the signal controllers use the same cycle length which is determined by their regional controller, *i.e.* the regional controller calculates the cycle length of its region for each unit optimization period whose minimum duration is $2\frac{1}{2}$ minutes. The cycle length is modified in increments or decrements of a few seconds. The cycle length of each region has its own preset upper and lower bounds. The objective of varying the cycle time in SCOOT is to ensure that the most heavily loaded junction operates at a specified

maximum degree of saturation (DS), which is the ratio of the effectively utilized green time to the total available green time. The maximum DS is specified at 90 % of the saturation limit. The computed cycle length can be doubled at any region if considerable amount of reduction in the delay is possible due to this doubling of the cycle length. Moreover, a single intersection can work at half the cycle time of the region if the cycle length is found to be very high for that intersection.

SCAT also divides the traffic network into regions; each region is controlled with one common cycle length as in SCOOT. The DS is also used in SCAT as one of most important factors in its computation. Data from detectors are used for the computation of the DS. If the DS in a phase is higher than the specified threshold then the green time is extended for that phase in the next cycle. The cycle length is computed by considering the link with the highest DS to reduce the volume in that intersection. The regional controller chooses the longest cycle length computed for the cycle length of the region.

b. Offset :

For each cycle in a region, the regional controller in SCOOT *calculates* the offset for each intersection based on information collected from the detectors at signal controllers. The purpose of this offset is to improve the overall traffic progressions on those streets which are immediately upstream or downstream of the intersection. The delay time and number of stops are used as the performance index measure.

In SCAT five internal offset plans, which refers to the offsets between each pair of intersections in a region, are provided to be *selected* on the basis of the traffic flow. The selection of an offset plan provides a coordination among all the intersections to benefit the flow. The offsets between different regions, which refer to the offsets between the intersections on streets linking two neighboring regions, are selected from five external

offset plans depending on traffic conditions.

c. Split :

A few seconds before a signal change at an intersection, the signal controller decides as to whether it is advantageous to make the split as scheduled or extend the current phase time within the cycle length. Any such split alteration made to the green phase duration in a cycle is temporary since during the next cycle a new split decision will be used. These split alterations in the plan actively modify the traffic flow.

In SCAT there are four precalculated green split plans for each intersection which specify the proportion of cycle to be allocated to each phase, usually some phases are specified in percentage of cycle time and some in seconds. As cycle length increases, phases nominated in seconds receive a reduced proportion of the cycle and the phases nominated in percentages receive a constant proportion of the cycle. The green split plan is selected upon the basis of the highest DS in the region.

2.2 Discussion

Although the above two systems provide efficient control compared to other fixed time systems, there are a few limitations. In the SCOOT system, flow into congested areas is not restricted even if such an area is detected, because the split decision is made based on the DS in the incoming traffic to the intersection. In other words, each split decision assumes that the outgoing traffic is taken care of by the downstream signal controller. Thus, if the downstream controller cannot handle the traffic because of the congestion, then the link will be flooded. This inability to restrict the flow can become quite a serious problem since the congestion might spread.

Although SCOOT can set a progressive flow in the network, it is quite likely that the

progression can be broken if surges in cross traffic for that progression is encountered. Moreover, if two major progressive flows cross at an intersection then the signal provides an equal amount of green time to both the flows. This means that the progressing traffic approaching the intersection gets more green time along its path until that intersection. However, at the intersection, two flows from both directions are treated equally with the same amount of green time, hence becoming a bottleneck. Therefore, the approaches to the intersection may get flooded.

In addition, if a frequent occurrence of surges in traffic flow is sensed, then the system does not respond quickly due to the limitation in modifying the cycle lengths after a minimum of $2\frac{1}{2}$ minutes. If an accident occurs in a link and/or the link is entirely blocked, the signal controller at the entry part to that link will continue to remain green to traffic into this approach. This is because the fatality of the link is not known to the signal controller. Hence there is no efficient cooperation between signal controllers in extreme situations.

As only certain detectors are specified as strategic detectors for major flows, it is impossible for the SCAT system to respond effectively to abnormal flows sensed in non-strategic detector locations. The optimal timing plans to be selected are calculated on an off-line method. As in the SCOOT system, the detection of accidents is also not possible and the restriction of traffic flow into such a link is not enforced.

The disadvantages of OPAC have not been investigated completely but are found to have similar problems such as not restricting traffic into congested areas, difficulties detecting accidents and time consuming optimization. However, the OPAC system seems to employ very effective control mechanisms. We intend to study the system to very deep details.

2.3 The ARTC System Model

We now describe the physical and logical structure of our system. Since the ARTC system is a traffic-responsive control system, it requires detectors to be placed in all approaches to an intersection. In the present design we use detectors embedded in each lane of an approach. The *entry detectors* are placed at the entry point of the lane. The entry detector keeps count of the number of cars entering the lane. The *exit detector* placed on the stop line of the lane at the intersection is used to keep count of the cars exiting the lane. The *optimization detector*, which is a non-locking detector, is placed midstream or in a strategic position to warn the controller of the queue extending over its limit.

The number of cars in the *"buffer"* (the space between the optimization detector and the exit detector) is estimated by using the data from the three detectors. The count of vehicles in the buffer is used for estimating the queue length so that timely action can be taken to release the queue before it overflows the buffer. It is also used for setting the offset time to change the lights. A signal controller, using the information from the detectors and by communication with neighboring controllers, makes its decisions on split, offset and cycle times.

The regional controller collects traffic flow data and timing plans from the signal controllers periodically. Using this data it observes the change in traffic flow patterns, especially increasing traffic volume patterns, and sends specific control commands to the signal controllers to stabilize the traffic flow and thus to prevent congestion. These regional controllers are interconnected to provide improved flow control between regions.

3 THE ALGORITHMS

In this section, we present algorithmic steps in the computation for the control protocol in the ARTC system. We explain the control protocol in stages. First, we present the protocol under the following assumptions. All the detectors and computers are reliable hence the data obtained from them are precise. The communication links between the signal controllers are in perfect condition so that they deliver messages accurately to signal controllers. No accident occurs and there are no parking lots between signal controllers in the system. A two-phase control is assumed to be used by the signal controllers at all intersections. We then extend the protocol to handle parking lots and accidents. In the subsequent section, we describe the function of regional controller, which provides further optimization of the congestion problem and progression for a steady traffic flow.

The signal controller in the ARTC system is designed to store data which reflects the history of the traffic flow collected from detectors, *i.e.* the number of cars detected during each unit period of time is kept so that the intersection can predict an arrival pattern using this data. This type of data storing technique, called Cyclic Flow Profile (CFP), is used in SCOOT [16]. The vertical axis represents the number of the vehicles and the horizontal axis represents time. The different peaks and lows in the histogram depict the size of the platoons. The location of the platoons in the lane can be estimated by the time. In addition, we assume the detectors can estimate the speed of the cars hence the controller knows the difference between light traffic and heavy traffic.

3.1 Basic Algorithm Structure

We now present the algorithm in a stepwise fashion. The stepwise algorithm gives an abstract view of the working of the distributed layer.

Step 1. A few seconds before a signal controller, SC_1 shown in Figure overleaf, changes green to two approaches it communicates with the downstream controllers, SC_2s, of those approaches into which the traffic will flow, to obtain load data which is used to compute the split and green length. For this, the expiration period of the phase has to be known, for which we employ the timer. A timer at SC_1 is used to indicate the start and end of a phase. Once a decision for split is taken, the timer is set to the time at which it should begin the phase and the time at which it should end. The impending expiration of the timer indicates that a phase is going to end. The reason for the above mentioned communication is to prevent SC_1 from flooding SC_2 by not allowing a higher flow than it can accept.

Step 2. The second step, which is executed by SC_2, is to send the requested data to SC_1.

- If the phase on the approach from SC_1 to SC_2 is currently in the red phase or cars are not moving even under the green phase, then SC_2 performs the following steps. First, SC_2 computes the number of cars waiting in the approach using the CFP data and the optimization detector (presence detector) as explained above. In addition, the flow rate in the approach during the previous green phase is calculated using the past CFP data. This flow rate is represented in terms of seconds. These data are used to form total load data of the approach. SC_2 then sends the load data to SC_1.

- If the approach is currently in the green phase and cars are moving, SC_2 then sends the current flow rate data to SC_1 as the load data. It is possible that there can be a standing queue when an approach has just changed green or is in the middle of a green phase. However, during the green phase if there are vehicles in the queue and there is no blockage in the intersection, they will exit the approach at an average flow rate which can be measured and will remain to do so as long as there are vehicles in that approach and the phase remains green.

According to the load information, SC_1 can adjust its outflow rate into that approach.

Step 3. SC_1 now computes the flow rate of the all its approaches and the queue lengths of any standing queues at the approaches. With the load data or flow rate data from the two SC_2s and the flow rate data and queue data for SC_1's approaches, the signal controller at SC_1 decides the duration of the forthcoming green phase.

Step 4. When the green phase for those approaches begin, SC_1 sends a message to both SC_2s informing the departure of platoons from SC_1.

Step 5. Upon arrival of this message, SC_2 now checks if it can adjust its split to give a progressive flow to this platoon.

- If the downstream approach from SC_1 to SC_2 is currently in a green phase, the load of the crossing approach at SC_2 is considered for extension of the current green phase. The travel time of vehicles is considered by providing information about the dimensions of the approach to the controller beforehand. This length of the approach can provide an approximate time of arrival for vehicles from neighboring intersections. When the load data is received from a neighboring controller, the time of arrival is checked to see if it can be accommodated in the green by extending the green within a certain prespecified bound, else the phase is changed for the crossing approaches which might be experiencing unnecessary delay. However, if the load of the crossing approach is comparatively negligible, then the green phase is extended.

- If the approach is currently in a red phase, the flow rate of the Crossing approach at SC_2 is considered for reduction of the green phase. If the flow rate of the crossing approach is considerably light, then the green phase is terminated earlier than it should.

Due to the variable duration of the phases, the cycle length is also variable. If we allow unlimited extension of the green phase to an approach, then it is possible that some cars which were not detected but are waiting in an approach with the red phase may suffer with a indefinitely prolonged red phase. In order not to abandon such cars, each phase has lower and upper bounds for its length.

3.2 Handling of Parking Lots and Accidents

Now let us relax the assumptions about the traffic conditions and observe the behavior of the distributed algorithm. The traffic conditions are now said to be realistic, with the presence of parking lots in the system and the occurrence of accidents. With these assumptions the model is slightly modified with the placement of detectors and stop lights in front of major parking lots. The major parking lots are the ones from which the flow of vehicles during their peak hour period is heavy enough to affect the pattern of flow to the adjacent intersections. The only function of the stop light at a parking lot is that it asks its adjacent signal controllers for allowance of its outflow.

The CFP does not always reflect the actual flow since the minor traffic flowing in side streets and parking lots may not be detected at the entry detectors. Thus, in the ARTC system, the load of an approach is adjusted to reflect such variations. For example, a statistical factor depending on time of day can be applied to the load information of an approach.

With the case of accidents it is slightly complicated to exactly determine if there has been an accident or not. If an accident occurred where only a lane or part of the approach is blocked (i.e. traffic is able to flow at a slow pace) then the signal controller can detect a reduction in the flow rate from the exit detectors but this does not necessarily confirm

the accident since this symptom may occur for the parking lot problem too. But if the accident is sufficiently major so as to block the entire approach then it can be detected because the vehicles which trigger the entry detector will not reach the exit detector and hence the controller can detect that the approach has been blocked. If the flow rate drops suddenly at the exit detector and the traffic load behind the exit detector grows rapidly, we can defensively assume that there has been some problem such as a car collision.

When an accident occurs on an approach in a progressive path, or the downstream signal controller of the approach gives more green time to a sudden increase in cross traffic, the capacity of the approach may be reduced suddenly. In this case, the approaches behind the problem causing approach in the path may get a snowball effect, which may congest the area undesirably. The downstream signal controller of the problem causing approach now sends a warning message to the upstream controller so that the upstream controller can reduce the influx to the approach spontaneously. In this way, any sudden reduction of the flow in an approach forces the downstream signal controller to inform the upstream controller.

3.3 The Regional Controller: Progression and Congestion Control

Changes in the flow pattern in the traffic network are detected by the regional controller by collecting data from all signal controllers in its region periodically. Although signal controllers cooperate with each other, it is not perfectly guaranteed that congestion does not occur. In handling the congestion problem, a more efficient way than to detect the congestion is to prevent the occurrence of congestion by maintaining a moderated traffic flow in the entire region. Thus, the regional controller periodically checks the traffic flow of the region, detects any traffic path which shows a critical increase, and then tries to reduce

that flow. Since the overall traffic flow is maintained at a moderate level, the congestion may not be propagated to the neighboring approaches, even if an approach is temporarily congested due to some problems.

A. Traffic Flow Information

A signal controller maintains the traffic flow information for incoming approaches and this information is periodically updated and collected at the regional controller for every time period δt. To indicate degree of traffic flow of an approach during a time period δt, we use *vehicle volume ratio*, VVR, which indicates the average ratio of the number of vehicles compared to the maximum number of vehicles that an approach can carry. We use the VVR, which can identify congested and loaded approaches, to detect flows in large streets which form a mesh structure.

At a given time T, the vehicle volume ratio of an approach is calculated as follows:

$$\text{Vehicle Volume Ratio(VVR)} = \frac{\sum_{t=T-\delta t}^{T} Number\ of\ Vehicles\ at\ time\ t}{\sum_{t=T-\delta t}^{T} Maximum\ Vehicle\ Capacity\ at\ time\ t}$$

Maximum vehicle capacity of an approach at a given time is determined by the length and width of the approach and the status factor which is determined according to the condition of the approach at that time. For this, a signal controller maintains the statistical data for the maximum vehicle capacity of the approaches according to the time of day. A signal controller can determine the number of vehicles in each of its approaches by using the CFP data. Thus, vehicle volume ratio can be easily calculated at each signal controller for the approaches.

We now use the VVR data to identify the heavy flow approaches by following the steps specified below. We define a range of VVR to be a critical zone, which implies that the traffic in the approach is dangerously high if its VVR value is in the critical zone. For safer

estimation we represent each discrete range of the VVR in levels. For example if we divide the VVR into 10 levels, then level 0 denotes that VVR lies between 0 and 0.1 and level 1 denotes that VVR lies between 0.1 and 0.2 and so on. An increase in the calculated level indicates an increase in traffic volume.

B. Selecting Potentially Critical Approach

An approach is said to be in the potentially critical state if its current VVR level is found to be higher than the critical level. Upon receiving a information collection request from the regional controller, the signal controller sends a reply message which carries the current VVRs of approaches incident to it.

C. Computation

The regional controller periodically collects information about the traffic levels from the signal controllers in its region. Each time this information is collected, the regional controller forms the traffic information graph denoted by TIG. Each node in this graph represents a controller of an intersection, and a directed edge between two nodes represents the approach between two intersections. Each edge in the graph represents the level of flow of the corresponding approach.

The TIG reflects the current traffic status in the region. In brief, the TIG at the regional is used to:

1. detect a part or parts of a region which carries heavy traffic flow and then
2. smoothly reduce the incoming flow into these parts so that the traffic flow stays at a moderate level over the entire region.

If the level of flow is in the critical zone then the edge is called a *critical* edge. When a set of such edges forms a path, then it is identified. A path formed by the critical edges

is called a *critical path*. When a critical path is shown on the TIG, the regional controller computes the duration of the green phase for the controllers in that path to reduce the flow along the path. The duration of the green phase of the controllers in that path reduces gradually from the head to tail. If multiple paths merge at an intersection, the influx into the downstream approach of the intersection must reflect the turning flows. In this way, the traffic flowing into the path is gradually reduced experiencing a moderately longer delay but avoiding imminent congestion.

The regional controller may participate in the offset timing planning for paths experiencing outstanding but not yet critical level of traffic flow. For example, if two outstanding paths cross each other, then the intersection at the crossing point may become a bottleneck. In this case, the offset timing plan computed for each path should depend on the split timing at the intersection. If the regional controller does not provide this service, then it is possible that signal controllers cannot provide a smooth flow to the traffic.

It may also be desirable to use the regional controller to calculate the offset timing plan if the path with outstanding flow is detected, hence the flow in the progressive path can be better and smoother.

One of the major functions of the information collection at the regional controller is provision of such information to the central control center. With our approach, we believe that the information can be effectively collected, displayed, and used for various purposes.

D. Detection of Critical Paths

The regional controller uses a list structure to represent the TIG. Each edge, with an unique identifier represented in the data structure used for the computation, is as follows:

Edge ID	Source	M	Destination	P1 set	P2 set

P1 set is the set of pointers to outgoing critical edges from the destination.

P2 set is the set of pointers to incoming critical edges from the source.

M represents the number of incoming critical edges to the source.

To detect the critical paths, the regional controller performs the following steps:

(1) All non-critical edges are removed.

(2) Select a critical edge.

Mark this edge as H_{edge}.

If no such edge is found, go to (4).

- (2.1) Visit the node incident on the tail of the edge.
- (2.2) Select one critical edge incident on the node.
- (2.3) Repeat (2.1) and (2.2) until a node with no critical incoming edge is visited.
- (2.4) (Now, the search continues in a reverse direction beginning at H_{edge}.)
  Visit node incident on the head of H_{edge}.
- (2.5) Select one critical edge incident on the node.
  If no such edge is found, go to (3).
- (2.6) Repeat (2.4) and (2.5) until a node with no critical outgoing edge is visited.
- (2.7) Store the node visited at the last step in the above process as the head of the path.

As an edge is visited, it is stored in a linked list, and then it is removed from the TIG.

(3) Go to step (2).

(The operational complexity of the above steps is $O(E)$, where E is the number of edges in TIG. Now, each linked list represents each critical path.)

(4) Visit the node with more than one incoming critical edge, N_1 and find the path that has N_1 as its head.

Merge the two lists.

(5) Search the lists for critical paths to find a node at a tail of a path whose identifier matches N_1.

(6) Repeat (4) and (5) until all critical paths are visited.

The computation described thus far takes $O(P)$ operations, where P is the number of critical paths in the graph. The detection of paths with outstanding but not yet critical traffic levels can be done in a similar manner.

4 CONTROLLER AND NETWORK DESIGN

Signal controllers interact with each other for live information exchange through the regional network in real-time. The regional controllers may have to perform heavy computations for congestion pattern detection and coordination of signal controllers. Thus, high performance RISC type processors and special purpose computing engines are expected to be used for the regional controllers. In this report, we focus on the signal controllers and the interconnection network for the signal controllers. Regional controllers will be studied at a later phase of this research.

4.1 Signal Controllers

Signal controllers are installed in the intersections of the surface streets to control the signal lights and vehicle sensors. They must perform computations correctly in the face of a hostile environment like high temperature and high humidity. Thus, it is necessary to use hardened microelectronic components in the signal controllers to avoid external environment damage.

A straightforward solution to the signal controller design is to use the hardened version of a popular microcomputer system such as the IBM PC/AT built with a better packing technology. Although hardened microprocessors have a higher noise margin than their regular counterparts, failures still occur randomly. The architecture of the hardened regular microprocessors is very inefficient in handling exceptional conditions like power disruption and transient noises that cause the software program to jump to some unpredictable position, *i.e.* the so called run-away process. Although it is possible to remedy some of these problems by the software design, the system design becomes very complex and cumbersome. Moreover, the system verification problem is also very difficult.

The other problem with existing microcomputer architectures is that they cannot prevent the malicious behavior of failed components. All microcomputer systems have a backplane bus as the communication medium of different modules in the system. For achieving a high system throughput, a module/board on an open bus standard like the VME bus or the Multibus bus is allowed to directly write into other module's memory. Thus, a failed module may grasp the bus completely so that no other module can access the bus, or the memory of a module may be maliciously altered by another failed one. Thus, not only the service time of signal controllers must be sustained to the maximum but also the operations of the failed signal controllers must be completely ceased by automatic power-off.

To sustain its continuous operation, redundant hardware must be added to the signal

controllers, and a modular system architecture based on a message exchanging bus is suggested. A minimal number of electrical signals like 8 to 16 data bits, a few bits of address and control signals are sufficient to implement the message exchanging bus. Since there is no cross module write operations on the bus, no module can alter other modules' memory contents. Since the required interconnection pins for the message exchanging bus are very small, we can easily add redundant bus sets between modules to improve the system reliability. Through the message exchanging bus, a module requests servicing by sending out a command message to another module. The receiver of the command message validates the message, executes the service request and then returns the results to the requester. Based on the modular system architecture, sufficient computing power is provided to the signal controller by carefully selected microprocessors. The main idea of the modular organization for the signal controllers is conveyed with a design example explained below.

The four major traffic control operations performed by the signal controllers are sensor reading, signal light switching, signal light timing computation and inter-node communication. For simplicity, each of the four functions is assumed to be implemented by one module in our design example, and thus, we have four modules called the input, output, computation, and communication modules, respectively, to perform the four operations indicated above.

The input module establishes the CFP and VVR for each approach of the intersection, and it informs the computation module of the updated traffic information with a message when a certain traffic condition is triggered. The computation module iteratively calculates the best signal setting based on the information received from the input module and the neighboring nodes. After a new signal setting is decided, the computation module issues a new command to the output module for switching the signal lights, and then, packs the

traffic information together with the newly decided traffic light setting into a packet for delivery by the communication module. To transmit the CFP information from the input module to the computation module for each round of computation, which is done, say once per second, a bandwidth of 10K byte-per-second is sufficient for the message exchanging bus design.

One microcontroller is suggested to be used in each module of the signal controller as the master of the module. The main advantage of using microcontrollers instead of regular microprocessors is that they have special built-in hardware features for fault handling. Consequently, the system chip count is significantly reduced, and thus the system reliability is greatly improved. Microcontrollers usually have the built-in hardware devices like watchdog timers, input/output ports, ROM/RAM, and power on/off operation modes, etc. to handle exceptional conditions. Thus, most commonly occurring failures, such as power failure and transient noises, etc. are directly detected by these hardware mechanisms. After the cause of disruptions cease, the application programs can be easily recovered based on the interrupt signals generated by these hardware mechanisms.

A large selection of microcontrollers are available for different computing requirements. For example, high performance microcontrollers such as the Intel MCS96 family that has a 16-bit arithmetic and logic unit and a large memory (ROM and RAM) are suitable for complex computations. On the other hand, microcontrollers that have only several hundred instructions and a few I/O ports are suitable for very simple operations, such as pre-timed signal switching.

A noteworthy advantage of using one microcontroller in each module is that the signal controller has a very good survivability rate under undesirable conditions. For example, when the input module of a signal controller fails, the computation module can make control

decisions solely based on the information received from its neighbors. On the other hand, if the computation module fails, the input module or communication module can make the traffic control decisions (based on a simpler alternative to the original algorithm) and command the output module. When both the input and computation modules fail, the communication module can still make control decisions based on the information received from its neighbors.

To achieve the maximum system reliability in the signal controllers, the fault-tolerance technique must be resilient to different types of faults related to the device fabrication technologies. Since most low level fault-tolerant logic designs like coding and self-testing methods have an imperfect fault coverage, we cannot rely on them to achieve a perfect fault coverage. Using massive redundancy to mask faults in every module has a perfect fault coverage, but the system cost is prohibitively high. Thus, a hybrid redundant system architecture is proposed to achieve a high system reliability at a reasonable cost. In the hybrid architecture, one module is triplicated, called a triad, to achieve the perfect fault coverage. Then less expensive fault-tolerance designs are applied to other modules. As a design example, the computation module is assumed to be triplicated in the rest of our discussion, and the system organization of the signal controller is given in Fig. 1 The triplicated modules serve as the system hard core so that the failure of cheaper fault-tolerance designs in other modules will be correctly detected. The basic idea of the triplicated redundancy design is to couple the three identical computation modules together and then have them to execute three identical copies of software in lockstep. Any fault in one module is detected and masked by the majority rule. A triplicated bus that has three identical bus sets must be used to support message passing and the communication between the triplicated and simplex modules. Each bus set is controlled by one of the three computation modules.

Every module sends/receives messages along one or all of the three buses. A failed bus set or computation module is retired, and the remaining two modules and bus sets continue to operate until another bus/module fails.

To access the three busses, each module is equipped with three transceivers of which one is attached to each bus. Each module can freely select any, or all, of the three buses at a time. When a simplex module (such as the input module) sends a message to the triplicated modules, the message is broadcast to all three modules on the three buses. When the three redundant modules have to transmit three (redundant) messages to other modules, each of the redundant modules transmits its message on one bus concurrently. Since each module has three receivers, a majority vote on the message will mask any single fault caused by the bus or the computer module. This clock synchronization scheme is illustrated in Fig. 5.

Less expensive fault tolerant designs such as redundant information, redundant computations, and the time-out techniques can be applied to the modules that are not triplicated. An example on the redundant information technique is that when an output module receives a command from the computation module, the output module does not accept the command unless the complemented form of the same command is received and verified. The redundant computation technique essentially essentially performs a computation with a known result before we actually execute a critical computation. The actual computation is carried out only when the result of the first computation matches the known answer.

The modules in the signal controllers will eventually fail even though they have different fault-tolerant designs. Since the signal controller has to instantaneously identify the failed module during the on-line traffic control, the *fail-stop* operation must be designed into every module of the signal controllers. Fail-stop means that all electrical signals of an electronic device are disabled after the device fails. When all spare parts (in a signal controller) are

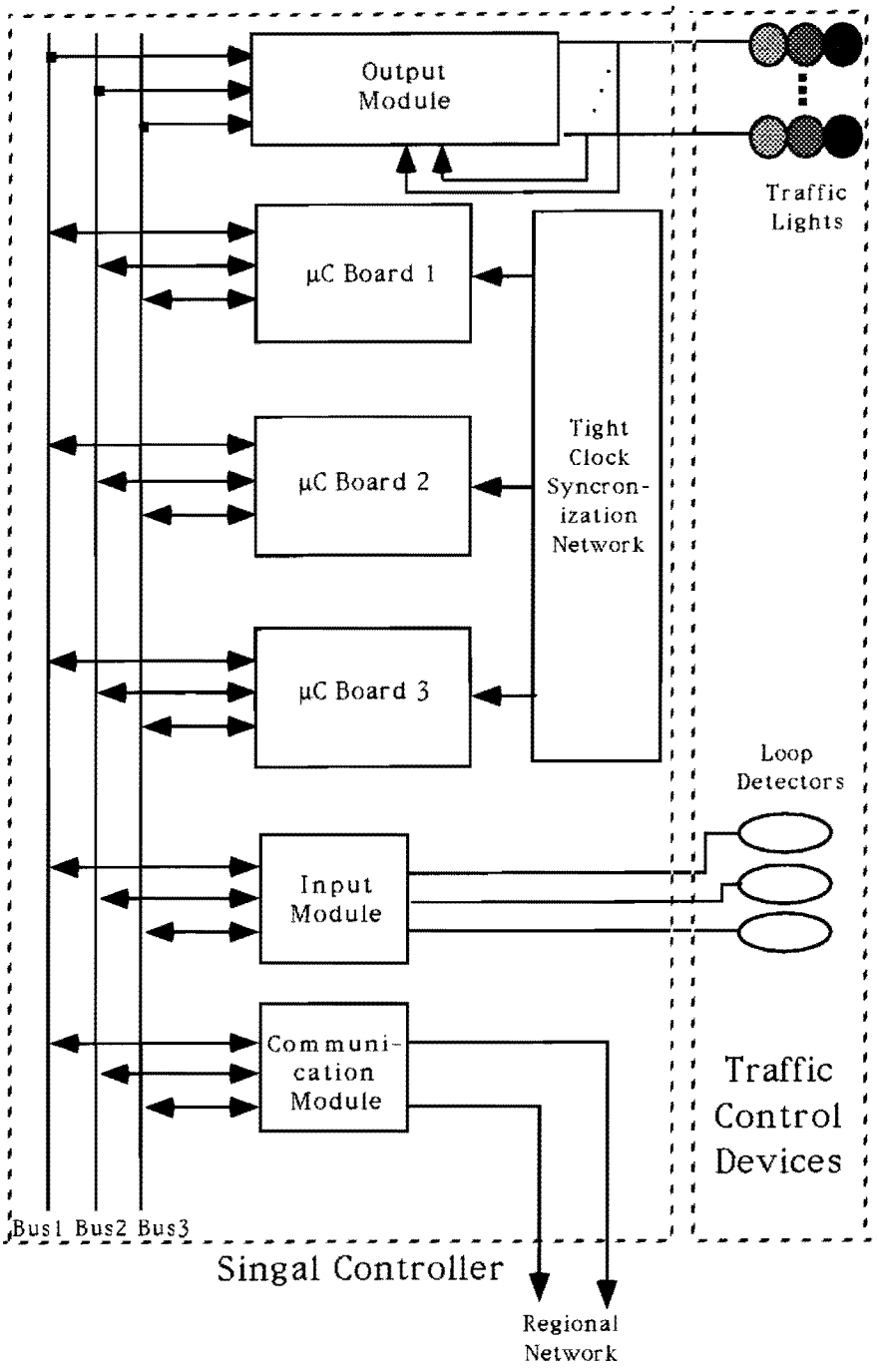

Figure 1: The System Organization of the Signal Controller.

exhausted, the power supply of the whole module is turned off automatically. The fail-stop operation of a functional unit can be implemented by using a watch-dog timer to disconnect the power supply of the functional unit if it fails to reset the watchdog timer periodically. The organization of different modules in the signal controller are explained in Appendix A.

4.2 Computer Networks

The three most important design factors of the computer networks for the advanced traffic management systems are the reliability, capacity, and cost of the networks. Although currently optical fiber is more expensive than other communication media, it has the highest reliability and extensibility in long run [17, 18]. Thus, the optical fiber is proposed to be used as the communication medium for the regional networks.

4.2.1 Regional network

In the distributed traffic control algorithm, adjacent signal controllers exchange live traffic information in real-time. Thus, the communication delay must not cause excessive distortion to the traffic information. For example, a vehicle traveling at the speed of 50 mph moves approximately 15 feet in 200 milliseconds. Thus, if the traffic information is delivered within 200 milliseconds, the traffic information distortion caused by such a communication delay is negligible for a street of 600 feet long. When one new traffic control decision is made every second and one computation iteration takes 100 milli-seconds, a network communication delay on the order of 200 milli-seconds will faithfully support execution of the distributed traffic control algorithms. Assuming that one block of street is approximately 600 feet long, and it takes five blocks to route a message from a source to its destination, the message must be processed and transmitted within 40 milli-seconds in each intermediate node. Based on

the the real-time traffic control requirement, we now estimate the minimum communication speed required for supporting the distributed algorithm execution.

The traveling speed of light is approximately one inch per nano-second (10^{-9} seconds). Since the optical signals in the cable are bounced along the surface of the cable, they are assumed to be transmitted at half of the straight line transmission speed. Thus, the total traveling time between the source and the destination of five blocks apart is around 72 micro-seconds (72×10^{-6} seconds). Thus, the signal traveling time on the network is negligible, and the communication delay is mainly determined by the data transmission speed, and the message processing times in the intermediate nodes.

Assuming that there are 10 bytes of data transmission in each round of information exchange, and it takes five milli-seconds to process one packet, and each intermediate node has five packets in its buffer, we must transmit the 80 bits of packet in three milli-seconds. Thus, the network transmission speed must be higher than $\frac{80}{3\times10^{-3}} \approx 2666$ bits per second.

This transmission speed lower bound is calculated solely based on data exchanges. It should be noted that the real-time constraint on inter-node communications must be satisfied even when some nodes/links on the network fail. Since packets must be rerouted in the face of the network failure, we must have an even higher transmission speed for the network.

An important issue in the packet rerouting is that packets rerouting may cause misordered packet transmissions that lead to an inconsistent system state [19, 20]. For example, under a sever weather condition, a signal controller fails to transmit a packet P_1 along a data link that has interfering intermittent noises, and thus, the signal controller reroutes P_1 along another path. After P_1 left the signal controller, the cause of the link interference ceases temporarily, and the next packet P_2 is routed along the previously interfered link

successfully. When P_1 and P_2 have the same destination, P_1 may arrive at its destination later than P_2 if P_1 is substantially delayed again in another node. Thus, an inconsistent system will be resulted if P_2 carries a command that is to overrule the command carried by P_1.

To avoid misordering of packets, we must establish a high resolution global time reference for the system so that aged packets are discarded when necessary. Furthermore, to avoid discarding too many packets, packets must be routed according to their priority. With the global time reference, we can easily define the priorities of packets on the network to best utilize the network by assigning the highest priority to packets that are closest to their expiration times. The packet scheduling problem becomes particularly important when more applications are accommodated into the ARTC system.

It is impossible to provide one single physical clock source for every node in the ARTC. Although the 60 Hz power lines have been used as the clock source in some existing traffic control systems, the 60 Hz resolution is inadequate for reliable and efficient coordination of signal/regional controllers. To provide an accurate global clock reference, the local clocks in signal controllers are adjusted to the master clock in the regional controller. Since the transmission time between two nodes is fairly fixed, the clock skew between the master and slave clocks is easily cancelled by a few rounds of message exchanges.

In addition to data transmission and scheduling, packet routing is the major computation burden to the communication module of the signal controllers. In general, the routing direction of a packet must be found from a routing table if the system does not have a systematic addressing scheme. On the other hand, with a systematic routing scheme, the next destination of a packet is calculated by using the addressing scheme. A systematic addressing scheme is also important for the expansibility of the network, because otherwise

we have to update all the routing tables in the network. Thus, a very flexible and efficient addressing scheme is developed for the regional network of the ARTC system.

Addressing Scheme and Fault-Tolerant Routing

To develop a systematic addressing scheme for the ARTC system, it should be noticed that most urban surface street systems have a fairly regular mesh structure with a small number of irregular branches. Thus to utilize this property, the road system is decomposed into two parts: the mesh network and the irregular branches. The horizontal and vertical coordinations of an intersection in the mesh are used as the address of the signal controller located at that intersection, and irregular branches are divided into *branch groups.* Signal controllers in the (extracted) mesh are connected by one network that is to be described later. Each node in a branch group is connected to two of its neighbors, and 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. The only constraint of the hierarchical addressing scheme is that all packets that must leave or enter a branch group have to pass its parent node.

Although a full mesh network can be directly used for interconnecting the mesh-like signal controllers, its cost is very high and utilization is very low. To reduce the network cost, some links in the full mesh network have to be removed. Thus, to maintain the internode real-time communication and fault-tolerance requirements, we build the network in two steps. First, vertical links are added to every node, and then, the horizontal links are added to the network one by one until the communication delay becomes sufficiently low. The resulting network is called a *sparse mesh network(SMN),* and an example of SMN is shown in Figure 2.

Each node in the SMN is connected to two or four nodes. The nodes that have two (four) adjacent neighbors are called the *chain nodes* (*cross nodes*), and several adjacent

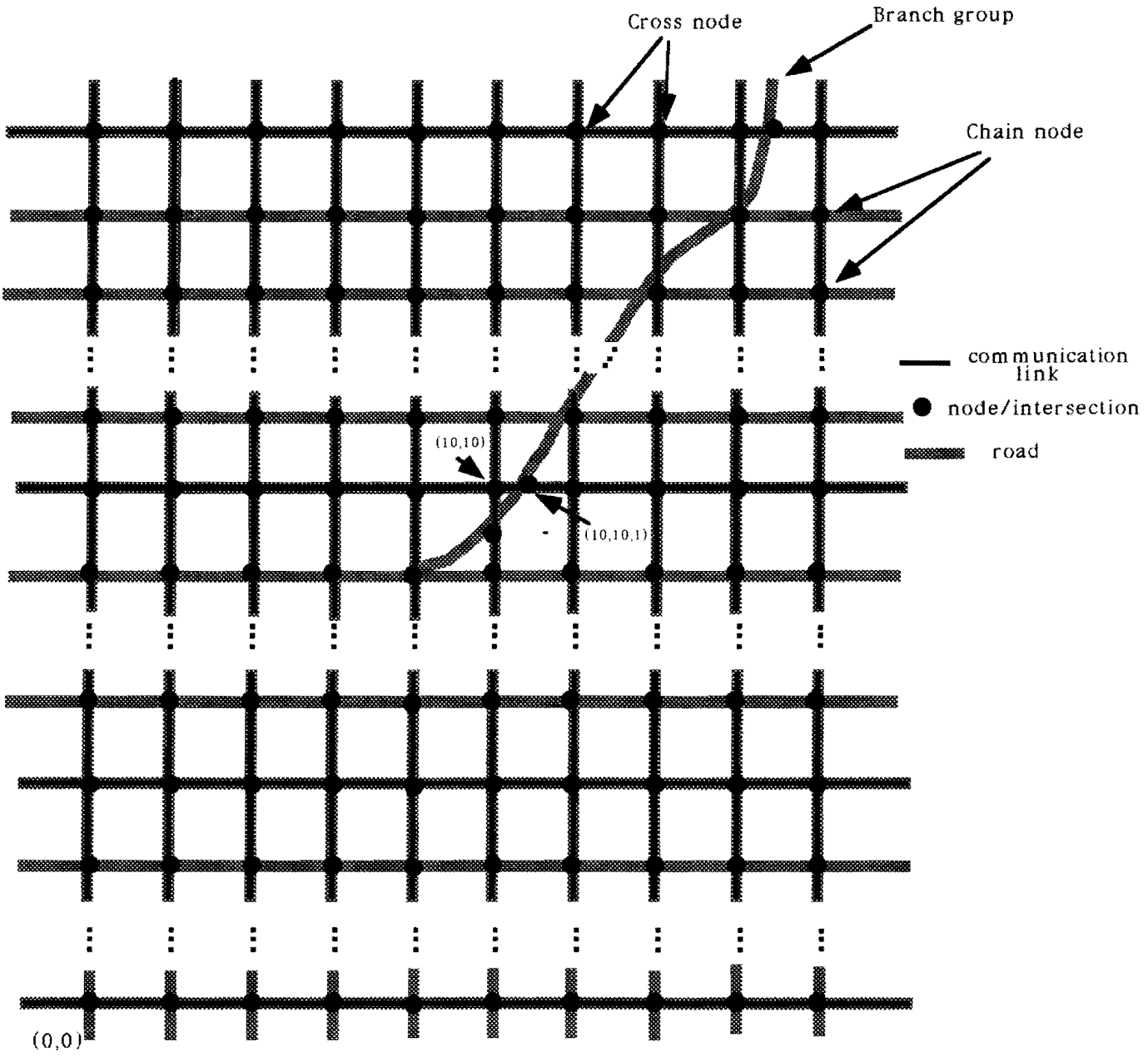

Figure 2: An Example of the Spare Mesh Network(SMN).

chain nodes may be located between adjacent cross nodes. Thus, when signal controllers are interconnected by an SMN, two adjacent signal controllers may have to exchange information by passing messages along some intermediate signal controllers.

A two-layer hierarchical address scheme is used to combine the mesh and the branch groups. The first layer of addressing space is reserved for nodes in the mesh, and the second layer of addressing space under a mesh node N_x is reserved for the branch group connected to N_x. Since the size of a branch group is usually fairly small, the routing problem is easily solved by adding a routing table to the nodes in the groups and the parents of the branch groups. When the destination of a packet is on the mesh, only the top-level layer address field is needed for packet routing. Otherwise, we use both layers of address fields for routing packets to nodes in branch groups.

The hierarchical addressing scheme has a very good expansibility. For example, several regional networks are integrated into a larger network by adding one more layer of addressing space on top of the regional network addressing scheme. If more subnetworks have to be added under one regional network, another layer of addressing space can be easily appended to the existing regional network addressing scheme.

In a full mesh network, packets are routed based on the relative positions of the source and destination. We first define the coordination of the lower-left corner of the mesh network as $(0,0)$. When a packet is in a node N_i with a coordination (X_i, Y_i), the node determines the next moving direction of the packet by calculating $DX_i = X_d^c - X_i$, and $DY_i = Y_d^c - Y_i$, where X_d^c and Y_d^c are the x and y coordinations of the destination node. The packet's optimal vertical routing direction is upward (downward) when $DY_i > 0$ ($DY_i < 0$), and the packet's optimal horizontal routing direction is rightward (leftward) when $DX_i > 0$ ($DX_i < 0$). The packet routing technique on the SMN is modified from that of the full

mesh network.

On the SMN, not all adjacent nodes have direct links, and thus the packet routing is divided into the micro and macro levels, respectively. At the micro-level routing, a packet in a chain node N_c is routed to its neighbor, which is in the opposite direction of the node that the packet came from if N_c is not the destination of the packet. The macro-level routing is performed by the cross nodes to find the direction of the next cross node for each packet (using the full mesh addressing scheme).

For example, when both the source and the destination of the packet are chain nodes but no direct link exists between the source and destination, we must route the packet through intermediate nodes. Since the network has the mesh addressing scheme, the relative position of the source and the destination is directly derived from their addresses. If the destination of the packet is on the same column as its source, the packet is routed to its destination in a few hops easily. Otherwise, the packet is first routed to N_c, which is one of the two cross nodes that are closest to the source node. Then, N_c calculates the cross node N_d that is on the same column as the destination node of the packet using the topology information of the SMN, and the moving direction of the packet is determined by N_c by treating all the cross nodes on the SMN as a full mesh. Since the same algorithm is executed in every node, the packet is guaranteed to be routed to N_d. From there, packet routing is straightforward.

The basic routing procedures mentioned above work efficiently when the network is fault-free. When a node or link in the network fails, the packet will be blocked in the node next to the failed node/link. Thus, the packet must be rerouted in order to reach its destination. A fault-tolerant packet routing technique, which is explained by the following example, that combines path blocking and packet rerouting solves this problem effectively.

In an SMN, a packet P_b is routed into one direction, say upward, by a cross node N_y,

and N_y is the last cross node that P_b passed. Due to a link failure, P_b is blocked in a chain node N_x before it can reach another cross node. Thus, N_x reroutes the packet P_b downward, after the blockage condition is piggybacked on P_b. When P_b is relayed back to N_y, N_y realizes that a link in the upward direction is blocked according to the blockage information piggybacked in P_b. Thus, N_y will reroute P_b into a new direction after the blockage information is removed from P_b. In the mean time, N_y does not allow any new packet to be routed upward unless the destination of the new packet is one of the chain nodes between N_x and N_y or the blockage condition is cleared.

A time-out parameter T is added to each packet, where T is the maximum number of hops that the packet can be routed, and the packet is discarded when T is decreased to 0. This fault-tolerant routing algorithm guarantees that no packet is bounced indefinitely, and it always finds a path to the packet's destination as long as the network is not partitioned. Since no extra information is required for the packet rerouting, the routing technique is very simple and efficient.

4.2.2 Network level fault-tolerance

Extensive hardware fault-tolerance features are to be built into the signal controllers. The fault-tolerance design in signal controllers emphasizes prolonginging the lifetime of the signal controllers and controlling the behavior of a totally or partially failed signal controller. Our signal controller design is expected to achieve over 95% fault coverage for its extensive and diverse fault-handling designs. It is extremely difficult and expensive to design a fault-tolerant computer that has a perfect fault coverage. Although we can increase the redundancy and intelligence of signal controllers to achieve an even higher degree of fault coverage, doing so with existing technology will increase the complexity and cost of the

signal controllers significantly. Thus, to minimize the possibility of undetected faults in the signal controllers, the network level fault-tolerance strategy is needed.

At the network level, the fault-tolerance strategy emphasizes early detection of failed signal controllers and prevention of the propagation of possible false information generated by the failed signal controllers. In a distributed computing system, computers can be centrally tested by a central controller or mutually tested by each other. Both approaches have some advantages and disadvantages, as explained below. The diagnosis problem is simple in the centralized testing approach, but the testing time is very long when a large number of computers need to be tested. Mutual testing is highly efficient, but the fault identification problem may become very complicated, because a failed tester cannot conclusively determine the functionality of its testee(s), as shown in Table 3.

tester	*testee*	*syndrome*
N	N	0
N	F	1
F	N	×
F	F	×

Table 3: Syndrome table for the fault-detection.

In this table, **F** denotes that the component is faulty and **N** denotes that the component is not faulty. The **1** (**0**) indicates that the fault is (not) detected and that (no) acknowledgement has been received by the tester. The × denotes that the result is unreliable and must not be used. Thus, to reduce the system testing time with reliable testing results, a hybrid testing approach that combines the distributed and hierarchical testing approaches is suggested.

That is, the tester-testee relation between signal controllers is first defined by a regional controller. Several testers are assigned to one testee and vice versa. Since a tester is also tested by other signal controllers, we will conclusively determine the failed nodes even when a failed tester gives unreliable testing results.

In the mutual testing process, a tester sends a testing message to its testee and waits for the response from the testee. After the testee receives a testing request, a fixed computation routine or component scanning routine is executed. Then, the testee sends the scanning/computation results as its response back to the tester. If the tester does not receive the response from its testee in a certain time period, it implies that either tester, testee, or the direct link between them has become faulty.

To test for a link failure, the tester node issues a packet to the testee node through a different route. If the testee node does acknowledge receiving the new link testing packet, it implies that the direct link between the tester and testee has failed. Otherwise, either the tester or the testee has failed. The testee reports all testing outcomes to the regional controller for further actions.

After the regional controller receives a complaint about the possible failure of one signal controller, the regional controller instructs the surrounding signal controllers of the suspected signal controllers to test the suspected failed nodes. Unless all the newly added testers are faulty, the suspected node is likely faulty if all the newly assigned tester nodes report that it has failed.

Once one failed signal controller is identified, the regional controller broadcasts a *warning message* to every node so that no malicious message from the failed node can be propagated to other nodes. The main idea of the fault-tolerant broadcast algorithm is that it generates a breadth-first spanning tree during the warning message broadcast, where the broadcast

message initiator is the root of the spanning tree. That is, after a node receives the broadcast message from one link, it relays the broadcast message to all its other links. Then, it discards all newly arrived broadcast messages. This technique guarantees that the distance from the broadcast root to the leaves is minimum, and each node receives the broadcasting message only once.

5 DISCUSSION AND FUTURE WORK

In this report, we have presented basic structure of the ARTC traffic controlling system which we believe will improve the traffic throughput considerably. With active communication between signal controllers, congestion prevention in traffic network can be effectively achieved, and progression service to the traffic can be optimized. Since it is very expensive to collect perfectly accurate traffic data, a high level control algorithm is added to the ARTC system. In the high level control algorithm, the ARTC system reduces the traffic gracefully before the traffic reaches a dangerous level, hence providing a congestion-free traffic control.

The performance of the ARTC system is yet to be evaluated. In our current design, only the outline of the decision parameters used in the control computation is specified. Thus, they have to be carefully studied and evaluated. In what follows, we briefly explain the work to be done in the future.

- Evaluation of the ARTC System Performance by Simulation and Computation Parameter Adjustments

  Simulation environment set-up for the performance evaluation of the ARTC system is to be done. In the simulation, the existing traffic system models are to be reviewed. The ARTC will be thoroughly compared with existing systems such as SCOOT, SCAT,

and OPAC, etc. The study on these successful systems in much more detail will help us identify any undiscovered concepts. Moreover, an intensive study of the systems which have not been successful will be done to identify the reasons for their failures. A simple demonstration of a graphical representation of a small network of roads will be given. A network wide demonstration will be given after the first phase demonstration is completed.

- Algorithm Elaboration and Refinement

  The algorithm at present is at a conceptual level with a few details being analyzed. The CFP nature of storing data from the vehicle detectors and using it for computation has to be researched. The precise method to adjust the duration of phase, extension of phase, turns, and defining upper and lower bounds for the phase is to be formalized. In order to adjust to the dynamic change in capacity a statistical factor has to be used and this factor has to be determined with further research.

- Fault Tolerant Algorithm Design

  The fault tolerant algorithm will be developed in the next phase. The performance of the algorithm has to be improved when certain hardware modules or software failures occur. The steps to take when a detector fails or the data is unreliable will be designed. The steps which should be followed when one or many signal controllers fail will also be given.

- Hardware Implementation and Testing

  It requires extensive effort and capital investment to implement a prototype signal controller, and this should be done step by step. In the first step, we plan to evaluate the computing power of some important microcontrollers. One or more evaluation

boards are to be tested. We will use the evaluation board to emulate functions of different modules of the signal controller to understand the speed of computation of the candidate microcontrollers. Then, inter-microcontroller communication will be experimented with, and a simple real-time signal controller without fault-tolerance features will be demonstrated.

The physical environmental constraints and current practice will be reviewed. Existing traffic light actuators should be adopted for the ARTC if they have a reliable field record. Statistics on the causes of failures for existing signal controllers have to be obtained for the fault-tolerance design of the signal controllers in the ARTC. Then, spare loops with embedded exercisers are to be added to existing loop detectors.

In the second stage, the fault-tolerant signal controller will be developed based on the features of the selected microcontroller. The specifications of the redundant bus to be used for such signal controller will be defined. The prototype emulates the signal controller located at an intersection operating with simulated detector information. This will reflect a realistic performance of the ARTC system.

- Operating Systems and Communication Network Design

  The operating system software to drive the designed controller is to be written, since the designed controller is not a general purpose machine. After the operating system is designed, the interconnection of those controllers will be designed and then tested. The communication protocol software design will follow.

Appendix A: The Frame Work of Signal Controller Design

Input Module

Standard loop detectors are used for the ARTC. Three loop detectors called the exit detector, optimization detector, and entry detector, respectively, are placed in each lane of the intersection, as illustrated in Fig. 3. To count the number of cars in each link of the intersection, an array of counters is needed. Another main component in the input module is the interrupt handler which handles the interrupts generated by the optimization detectors. The input module transforms the sensor signals into CFP and VVR for each approach. Thus, a high-performance microcontroller has to be used in the input module.

The other main function of the input module is to detect failed loop detectors. To improve the reliability of the loop detectors, redundant loop detectors with embedded loop exercisers are suggested for the ARTC. A redundant loop detector basically consists of two loop detectors connected in parallel. The spare loop immediately starts to detect vehicles once the primary loop failed. The loop exerciser is essentially an electrical magnet that generates a magnetic field to stimulate the loop(s) periodically. Thus, the failure of a loop is easily detected when the loop does not respond to the loop exerciser.

Output Module

The main function of the output module is to convert the small computer signals into large power signals for turning on and off traffic lights. Without a proper signal isolation, the large power signal may penetrate into the microelectronic components through a failed power control device and destroy the whole signal controller. Thus, all devices related to the power signal control should be isolated from the rest of the signal controller by the optical couplers.

During normal operations, the output module makes signal light switching based on

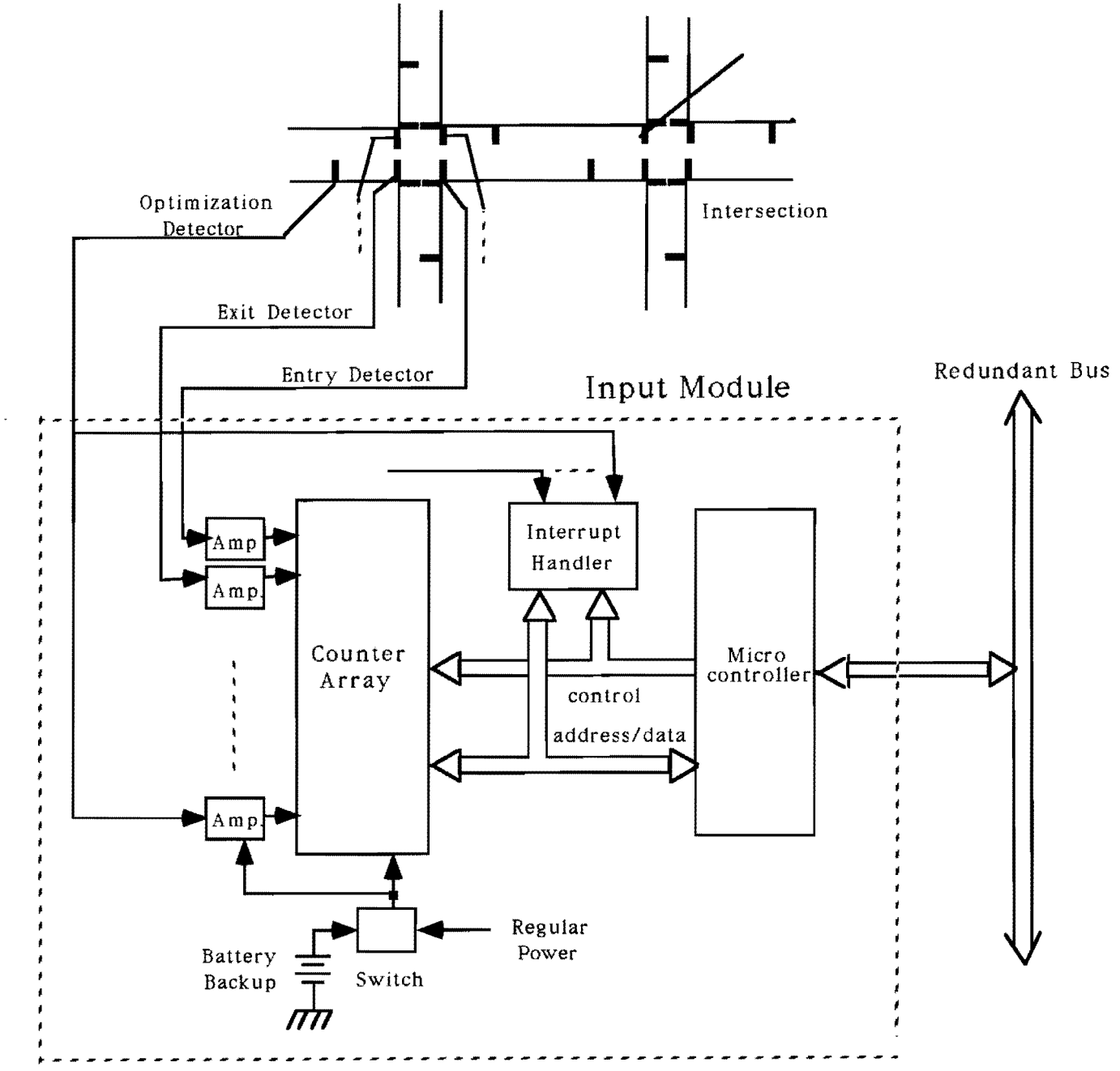

Figure 3: The Input Module of the Signal Controller.

commands received from the computation or communication modules. Since the signal light switching does not happen frequently, we have a sufficient time for checking possible errors in the received commands by a software technique like message coding. A faulty command is discarded, and a negative acknowledgement is sent back to the sender.

When the traffic system is controlled by the adaptive control algorithm, the cycle times of the intersection vary. To avoid possible mistakes in the program, the output module keeps track of the duration of the current state. The computation (communication) module must issue commands periodically to the output module even though the signal setting is not changed. The output module first issues a warning command to the computation module if it did not receive commands after a certain period. If the computation module does not respond to the warning message, the output module enters a degraded operation mode and asks for the attention from other modules by sending a panic message.

For maximizing its service time, the output module as illustrated in Fig. 4 is designed to have at least three different signal switching modes, each of which is controlled by a different hardware mechanism. During normal operation, the output module is controlled by a microcontroller μ_1, and the output module is in the fully adaptive control mode. That is, the output module switches signal lights according to the commands issued by the local or remote computation modules. After μ_1 fails, another (small) microcontroller μ_2 takes over the switching function based on the timing plane(s) set by the μ_1, *i.e.* the local pretimed operation mode. Finally, after both μ_1 and μ_2 fails, the output model enters the red/amber flashing mode that is controlled by a small timer.

Computation Module

The main function of a computation module is to execute the distributed traffic control algorithm with input data from the input module and the neighboring nodes. To provide

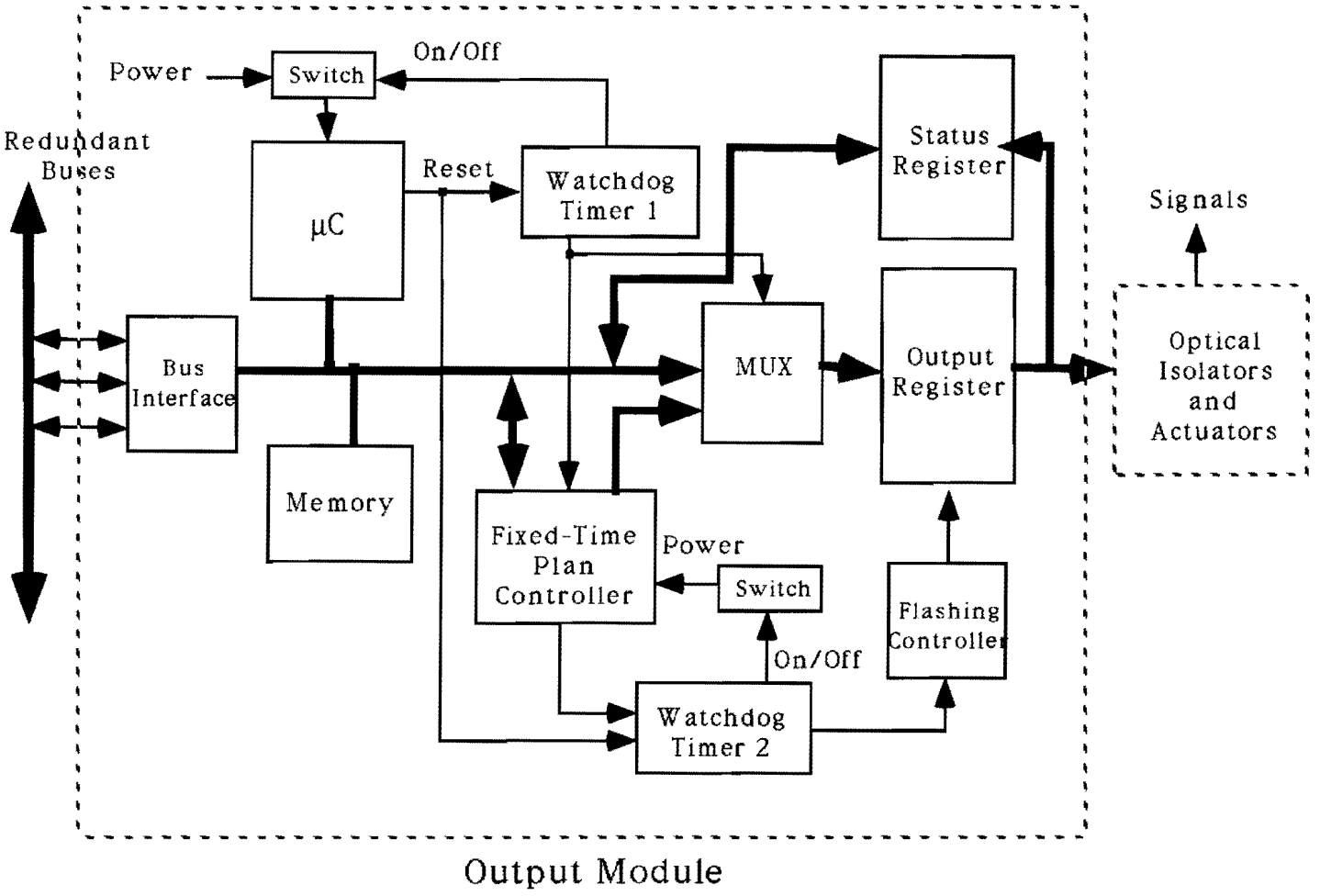

Figure 4: The Output Module of the Signal Controller.

sufficient computing power and fault handling features, 16-bit microcontroller chips like the INTEL MCS96 series[21] or other similar products can be used. The MCS96 microcontroller chip runs at 12 Mhz, and it has internal PROM, RAM, programmable timers/counters, internal/external interrupt handler, I/O ports, and watchdog timers as shown in Fig. 6. Thus, with some more external memory, large programs can be easily accommodated into the computation module with only a few chips.

The traffic control strategies may evolve with time. Thus, a signal controller completely designed with the ROM or PROM makes the system upgrading and maintenance difficult and costly. A possible solution to the system upgrading problem is to add remote memory updating capacity to the signal controllers. That is, the PROM in the microcontroller only stores the core system management programs and then the application programs are stored in the external flash memory. The flash memory is not volatile, and its contents are erasable and programmable with certain defined sequences. Thus, new programs can be easily loaded and updated during on-line operation after the microtroller receives the upgrading commands and new programs from the regional controller.

When the computation module is triplicated, the redundant computation modules execute programs in lockstep by tightly synchronizing the clocks of the three modules, as shown in Fig. 5. Since all the redundant modules run at the same speed, and all of them have the same inputs, they generate identical outputs simultaneously. The outputs are transmitted on the triplicated buses simultaneously to other modules.

Communication Module

The communication module plays a crucial role in the signal controller design. Depending on the communication speed and the packet size, the communication module design can be very simple or very complex. Since most of the packets on the communication network

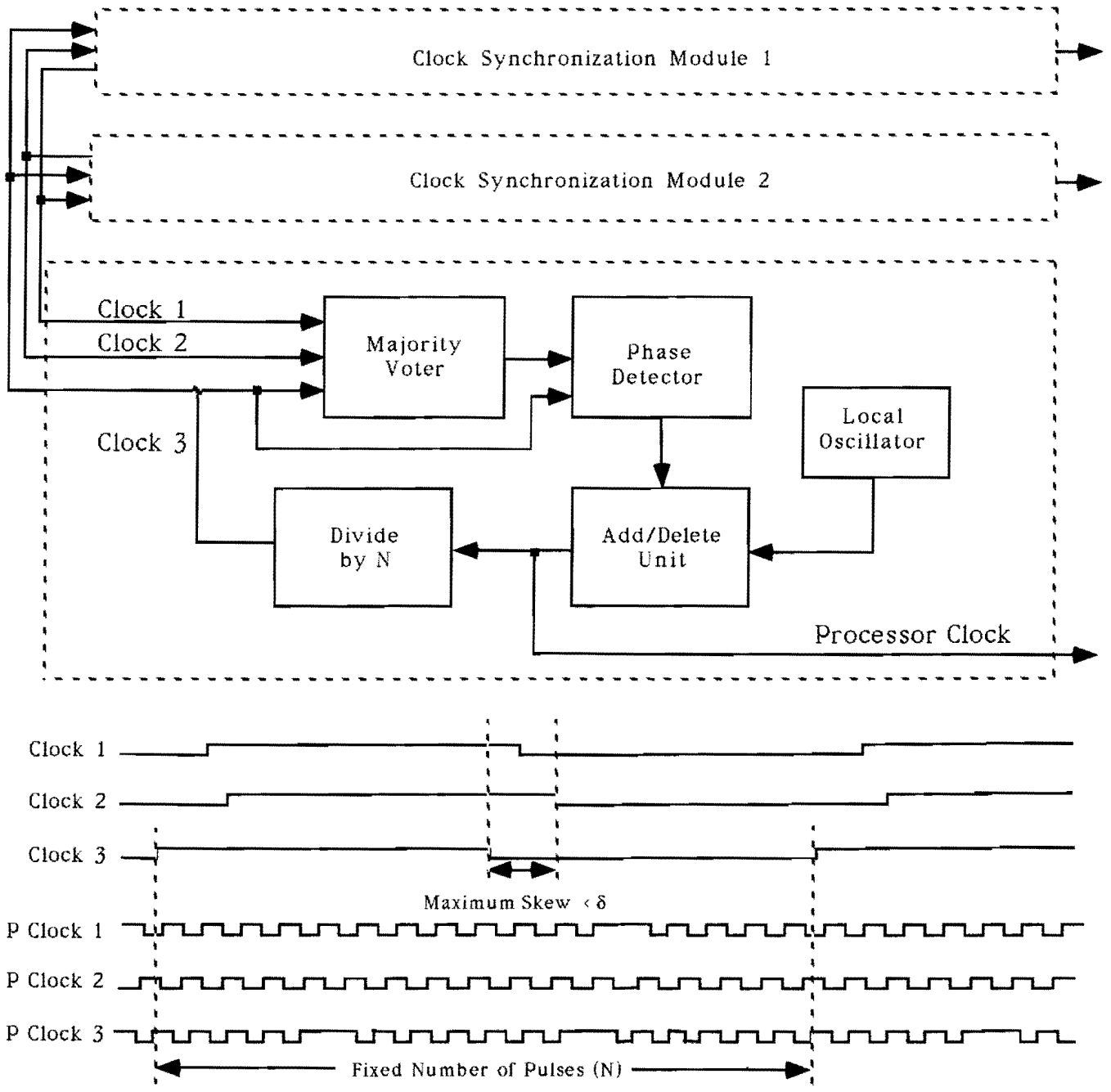

Figure 5: The Tight Clock Synchronization Network.

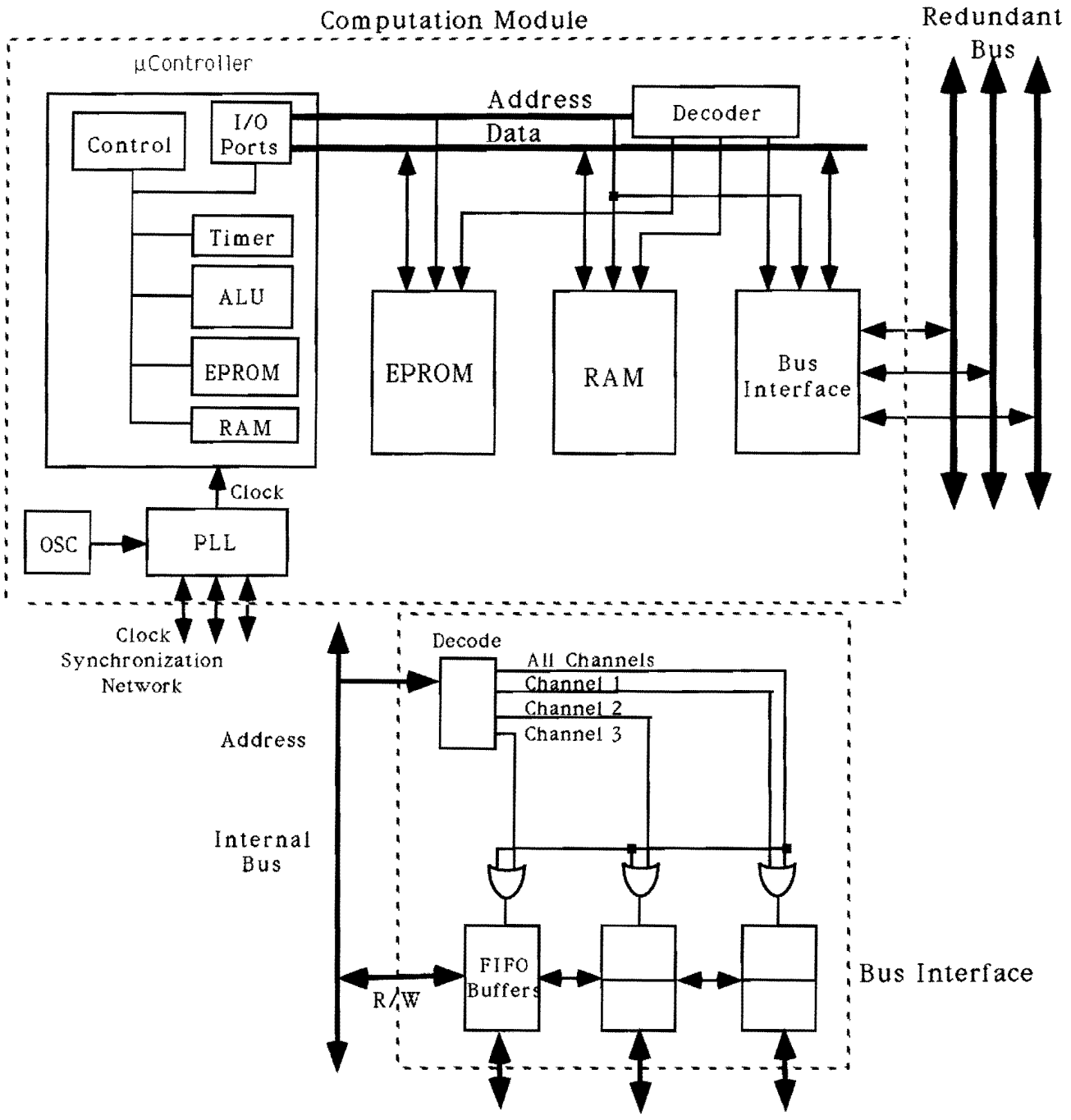

Figure 6: The Computation Module of the Signal Controller.

have real-time constraints, the packet routing should be done by a custom communication processor. Otherwise, the communication module may become the performance bottleneck of the ARTC system. The communication processor also handles link-level protocols of messages transmitted to/from the regional network. In addition to the communication protocol handling, packet processing is also a very important task that has to handle the large number of interrupts generated by the departures/arrivals of packets.

Before the optical fiber bus technology is available, we must use the point-to-point interconnection for the optical fiber networks. Thus, the network operation will be greatly affected when the communication modules of signal controllers fail. To maintain the maximum network connectivity, a relay is added to the transceivers of the communication module. Thus, messages that must pass a failed node are relayed to the next adjacent node without packet loss. Fig. 7 illustrates the communication module.

The signal controller design presented up to now is based on the hypothesis that the hardware components are moderately expensive and the computation module is the only module that has a sufficient computing power. It is expected that the price of microelectronic devices will steadily fall in the future. Since the main fault-tolerance technique is fairly independent of the technology, our current design can be easily upgraded in the future for even better performance and reliability.

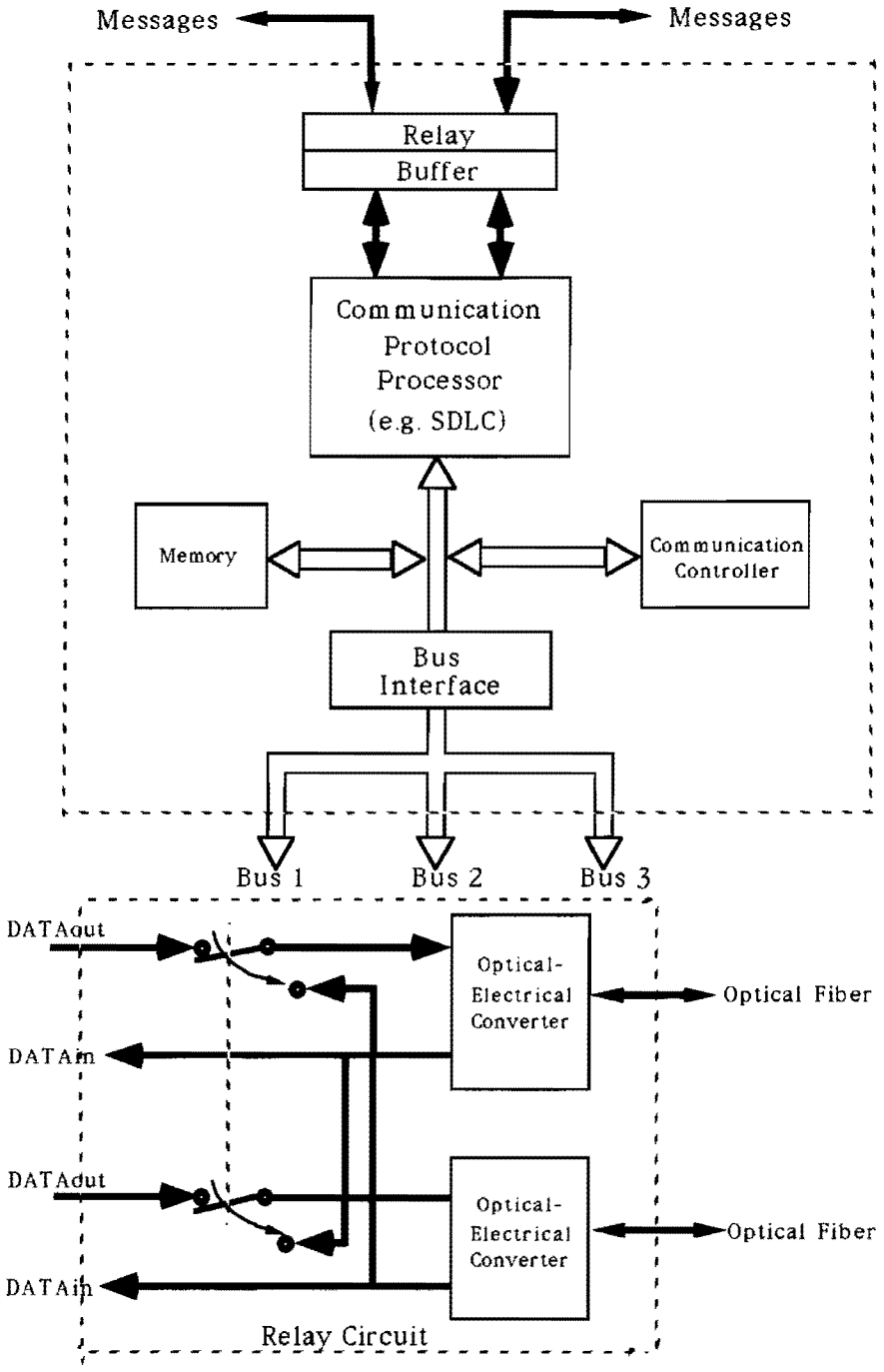

Figure 7: The Communication Module of the Signal Controller.

References

[1] 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.

[2] 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.

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

[4] F. V. Webster, "Traffic signal setting," *Road Research Technical Paper, Road Research Laboratory*, no. 39, pp. 1–44, 1958.

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

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

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

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

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

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

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

[12] J. Walmsley, "The practical implementation of SCOOT traffic control systems," *International Conference: Road Traffic Signalling*, no. 207, pp. 197–202, 30 March - 1 April. 1982.

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

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

[15] J. Y. K. Luk, A. G. Sims, and P. R. Lowrie, "SCATS - application and field comparison with a transyt optimised fixed time system," *International Conference: Road Traffic Signalling*, no. 207, pp. 71–74, 30 March - 1 April. 1982.

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

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

[18] U. Black, *Data Networks, Concepts, Theory, and Practice*, Prentice-Hall Inc., Englewood Cliffs, 1987.

[19] W. Stallings, *Data and Computer Communications*, Macmillan Publishing Comp., New York, Collier Macmillan Publishers, London, 1985.

[20] M. Schwartz, *Telecommunication Networks, Protocol, Modelling and Analysis*, Addison-Wesley Publishing Company, 1987.

[21] *16-Bit Embedded Controllers*, INTEL Corp., 1990.