

MN DEPT OF TRANSPORTATION



3 0314 00023 8708

SOTA

**CENTER FOR  
TRANSPORTATION  
STUDIES**

994004

ITS INSTITUTE

**DEVELOPMENT OF TRAFFIC  
SIMULATION LABORATORY  
FOR DESIGN PLANNING AND  
TRAFFIC OPERATIONS  
(PHASE I)**

**Michalopoulos, Sommers, Reyhout, Kota,  
and Kwon  
Department of Civil Engineering**

CTS  
HE  
336  
.T7  
D483  
1996



1. Report No. CTS 97-11	2.	3. Recipient's Accession No.	
4. Title and Subtitle  Development of Traffic Simulation Laboratory For Design Planning and Traffic Operations (Phase)		5. Report Date  December 1996	
		6.	
7. Author(s)  Michalopoulos, Sommers, Reynhout, Kota, and Kwon		8. Performing Organization Report No.	
9. Performing Organization Name and Address  Center for Transportation Studies ITS Institute Program 200 Transportation and Safety Building 511 Washington Avenue SE University of Minnesota, Minneapolis, MN 55455-0220		10. Project/Task/Work Unit No.	
		11. Contract (C) or Grant (G) No.  (C) (G)	
12. Sponsoring Organization Name and Address  Center for Transportation Studies ITS Institute Program 200 Transportation and Safety Building 511 Washington Avenue SE University of Minnesota, Minneapolis, MN 55455-0220		13. Type of Report and Period Covered  Final Report 1995	
		14. Sponsoring Agency Code	
15. Supplementary Notes			
16. Abstract (Limit: 200 words)  The key element in improving traffic operations and performing effective real time management is the ability to assess the effectiveness of the various alternatives prior to implementation. Simulation methods have long been recognized as the most effective tool for such an analysis, and various simulators have been developed by different agencies for freeway and arterial networks. While there have been individual tests of each software by various agencies, no comprehensive effort has been made to quantify and evaluate the performance of each model. The major difficulty includes the lack of detailed real data and the time-consuming manual effort to prepare input information for each software. To be sure there is no user-interface developed to data that can generate input files for different simulators with a common set of data. Recent installation of the state-of-the-art video detection systems in the I-394 freeway provides a unique opportunity to develop a comprehensive freeway database that can be used to create various test cases with detailed traffic database that can be used to create various test cases with detailed traffic information. Further, the I-494 Integrated Corridor Traffic Management project being conducted by MN/DOT will be able to provide valuable corridor traffic data which can be used for evaluating freeway/arterial network models. By evaluating existing traffic simulation models the advantages/disadvantages of each model can be identified. Based on the evaluation results a comprehensive modeling approach for freeway/arterial networks can be developed. The ultimate goal of this research is to develop a traffic simulation laboratory where various roadway design/operational alternatives can be evaluated with traffic simulators under an integrated database-simulation environment. The proposed research, Phase I, will evaluate existing freeway simulation models with detailed traffic data to be collected using machine-vision an loop detection systems.			
17. Document Analysis/Descriptors  Variable Message Signs Work Zones		18. Availability Statement  No restrictions. Document available from: National Technical Information Services, Springfield, Virginia 22161	
19. Security Class (this report)  Unclassified	20. Security Class (this page)  Unclassified	21. No. of Pages	22. Price



**DEVELOPMENT OF TRAFFIC SIMULATION LABORATORY FOR DESIGN  
PLANNING AND TRAFFIC OPERATIONS (PHASE I)**

**Final Report**

Prepared by

Dr. Panos Michalopoulos  
Kevin Sommers  
Kenneth Reynhout  
Ramesh Kota  
Department of Civil Engineering  
University of Minnesota  
Minneapolis, MN 55455

Dr. Eil Kwon  
ITS Institute, Center for Transportation Studies  
University of Minnesota  
Minneapolis, MN 55455

**December 1996**

Published by

Center for Transportation Studies  
University of Minnesota  
Minneapolis, MN 55455



## Table of Contents

<b>Executive Summary</b>	i
<b>1. Introduction</b>	1
1.1 Problem Statement	1
1.2 Research Objectives	1
1.3 Report Organization	2
<b>2. Selection and Overview of Freeway Simulation Models</b>	3
2.1 Selection of Simulation Models for Evaluation	3
2.2 FRESIM	4
2.3 AIMSUN2	8
2.4 FREFLO	11
2.5 FREQ11	14
2.6 KRONOS	17
2.7 INTEGRATION	19
2.8 Summary of Simulation Models	24
<b>3. Development of Model Evaluation Plan</b>	29
3.1 Development of Database Structure	29
3.2 Development of Quantitative Test Cases	30
3.3 Development of Qualitative Test Cases	37
3.4 Calibration of Model Parameters	42
3.5 Input file Preparation	45
<b>4. Evaluation of Freeway Simulation Models</b>	47
4.1 Quantitative Test Cases	47
4.2 Qualitative Test Cases	56
<b>5. Development of a Framework for Automatic Simulation Environment</b>	64
5.1 Introduction	64
5.2 Design Considerations	65
5.3 Prototype Development	90
<b>6. Improvement and Support for the Minnesota Freeway Simulation Model</b>	107
6.1 Model Improvements	107
6.2 Support for Model Implementation	111
<b>7. Conclusions and Future Research Needs</b>	117
<b>References</b>	119
<b>Appendixes</b>	
Appendix A: Quantitative Test Cases	A-1
Appendix B: Qualitative Test Cases	B-1
Appendix C: Source Code for Prototype Automatic Simulation Environment	C-1

## List of Tables

Table 2.1 : Selected Simulation Models and Their Tested Software Release Versions	3
Table 2.2 : Simulation Model Computer Requirements and General Information	24
Table 2.3 : Simulation Model Capabilities	25
Table 2.4 : Simulation Model Inputs	26
Table 2.5 : Simulation Model Parameters	27
Table 2.6 : Simulation Run Times in Seconds for Selected Test Cases Using a Pentium Personal Computer Running at 90 Megahertz	28
Table 2.7 : Simulation Model Outputs	28
Table 3.1 : Data Entry Times in Work Days for the Quantitative Test Cases	45
Table 4.1 : Mean Percent Difference for I-494wb South of Twin Cities International Airport	47
Table 4.2 : Mean Percent Difference for I-494wb Through Southwest Minneapolis	50
Table 4.3 : Mean Percent Difference for I-94eb Through the University of Minnesota Ramp Reconstruction	53
Table 4.4 : Spillback Lengths and Release Times for I-494wb at the TH-100 Cloverleaf	57
Table 4.5 : Difference Between Upstream Flow Rate and Downstream Flow Rate (vph) at the Beginning of the I-494wb Test Section (600 ft to 5180 ft)	58
Table 4.6 : Difference Between Upstream Flow Rate and Downstream Flow Rate (vph) at the Midsection of the I-494wb Test Section (5980 ft to 11090 ft)	58
Table 4.7 : Difference Between Upstream Flow Rate and Downstream Flow Rate (vph) at the End of the I-494wb Test Section (11090 ft to 16115 ft)	59
Table 4.8 : Spillback Lengths and Release Times for I-35w Southbound Through Burnsville at Lane Drop Sections	59
Table 4.9 : Spillback Lengths and Release Times for I-35w Southbound Through Burnsville at Incident Section	60
Table 4.10 : Spillback Lengths and Release Times for Multiple Lane Drop Freeway Lane Drop Sections	60
Table 4.11 : Spillback Lengths and Release Times for Multiple Lane Drop Freeway Incident Section	60
Table 5.1 : GETRAM Design Description of a Section	97
Table 5.2 : GETRAM Design Description of a Node	99



## List of Figures

Figure 3.1 : Freeway Geometrics for Quantitative Test Case 1 ~ I-494 Westbound South of the Twin Cities International Airport	33
Figure 3.2 : Freeway Geometrics for Quantitative Test Case 2 ~ I-494 Westbound Through Southwest Minneapolis	34
Figure 3.3 : Freeway Geometrics for Quantitative Test Case 3 ~ I-94 Eastbound During the University of Minnesota Ramp Reconstruction	36
Figure 3.4 : Freeway Geometrics for Qualitative Test Case 1 ~ I-494 Westbound at the TH-100 Cloverleaf	39
Figure 3.5 : Freeway Geometrics for Qualitative Test Case 2 ~ I-35W Southbound Through Burnsville	40
Figure 3.6 : Freeway Geometrics for Qualitative Test Case 3 ~ Multiple Lane Drop Freeway Section	41
Figure 4.1 : Fresim Results Compared to Collected Flow Data at DS 183	48
Figure 4.2 : Aimsun2Results Compared to Collected Flow Data at DS 183	48
Figure 4.3 : Freflo Results Compared to Collected Flow Data at DS 183	48
Figure 4.4 : Freq11 Results Compared to Collected Flow Data at DS 183	49
Figure 4.5 : Kronos Results Compared to Collected Flow Data at DS 183	49
Figure 4.6 : Integration Results Compared to Collected Flow Data at DS 183	49
Figure 4.7 : Fresim Results Compared to Collected Flow Data at DS 515	51
Figure 4.8 : Aimsun2Results Compared to Collected Flow Data at DS 515	51
Figure 4.9 : Freflo Results Compared to Collected Flow Data at DS 515	51
Figure 4.10 : Freq11 Results Compared to Collected Flow Data at DS 515	52
Figure 4.11 : Kronos Results Compared to Collected Flow Data at DS 515	52
Figure 4.12 : Integration Results Compared to Collected Flow Data at DS 515	52
Figure 4.13 : Fresim Results Compared to Collected Flow Data at DS 467	54
Figure 4.14 : Aimsun2Results Compared to Collected Flow Data at DS 467	54
Figure 4.15 : Freflo Results Compared to Collected Flow Data at DS 467	54
Figure 4.16 : Freq11 Results Compared to Collected Flow Data at DS 467	55
Figure 4.17 : Kronos Results Compared to Collected Flow Data at DS 467	55
Figure 4.18 : Integration Results Compared to Collected Flow Data at DS 467	55
Figure 5.1 : Basic Structure of an Automated Freeway Simulation Environment	66
Figure 5.2 : Interstate 394 Laboratory Cameras	68
Figure 5.3 : Client-Server Architecture	75
Figure 5.4 : Data Separation Tree	77
Figure 5.5 : Example Loop Detector Station Report	78

## List of Figures

Figure 5.6 : Separating Sensor Data by Time	81
Figure 5.7 : Opening of Freeway Network File	84
Figure 5.8 : Sub-Network Selection	85
Figure 5.9 : Detailed Graphical Display	86
Figure 5.10 : Loading Sensor Data	86
Figure 5.11 : Choosing Sensor Data Storage Unit	87
Figure 5.12 : Sensor Data Query	88
Figure 5.13 : Sensor Locations in Sub-Network	88
Figure 5.14 : Starting a Simulation Run	89
Figure 5.15 : GETRAM Functional Design	92
Figure 5.16 : AIMSUN2 Simulation Process	93
Figure 5.17 : KRONOS Simulation Process	94
Figure 5.18 : GETRAM Geometry Structure	95
Figure 5.19 : GETRAM Section	96
Figure 5.20 : GETRAM Juncture	98
Figure 5.21 : GETRAM Junction	99
Figure 5.22 : KRONOS Geometry Structure	100
Figure 5.23 : Kronos Segment Types	102
Figure 5.24 : Translator's Main Program Structure	105
Figure 5.25 : TDTranslateToKronos Function Structure	106

## EXECUTIVE SUMMARY

A key element in improving traffic operations and performing effective real time management is the ability to assess the effectiveness of various alternatives prior to implementation. Simulation methods have long been recognized as the most effective tool for such an analysis and various simulators have been developed by different agencies for freeway and/or arterial networks. While there have been individual tests of each software by various agencies, no comprehensive effort has been made to quantify and evaluate the performance of each model. The major difficulty includes the lack of detailed real data and the time-consuming manual effort to prepare input information for each software. To be sure, there is no user-interface developed to date that can generate input files for different simulators with a common set of data. Developing an integrated traffic analysis environment, where data processing, simulation and output analysis can be automated as efficiently as possible, is of critical importance in improving traffic management and operations.

In this research, the six well-known simulators for freeways are evaluated with a common data set. These include FREFLO, AIMSUN, KRONOS, FRESIM, INTEGRATION and FREQ. First, a systematic testing plan was developed with real and hypothetical data to evaluate the quantitative as well as qualitative behavior of each model. A total of three quantitative cases were developed with the real traffic data collected from the Twin Cities freeway network, while six qualitative cases were developed. The test results show that the mean percentage difference between measured and estimated data in terms of five minute volume are less than 10 percent for most of models tested in this research. However, the lack of an automatic, optimal model calibration procedure and the unavailability of speed measurements make it very difficult to compare the performance of each model. In general, the qualitative behavior of all the models showed the expected trend.

In addition to evaluating the freeway models, a basic framework for an automatic simulation environment was also developed in this research. The main objective of the automatic simulation environment is to automate the time consuming input data preparation process, so that traffic engineers can spend more time in analyzing data rather than preparing input files for simulation. A prototype system using the GETRAM software was also developed and the basic concept of automatic simulation was demonstrated.

Future research needs include the development of a comprehensive data sets with various geometric and traffic conditions including speed measurements, an efficient procedure to test the quality of the data set, and an automatic model calibration system that can determine the best set of the parameters for each model for a given geometric and traffic conditions. Further, the automatic simulation system where data processing, input file preparation and analysis of the data can be efficiently performed in an integrated environment needs to be developed.

# 1. INTRODUCTION

## 1.1 Problem Statement

A key element in improving traffic operations and performing effective real time management is the ability to assess the effectiveness of various alternatives prior to implementation. Simulation methods have long been recognized as the most effective tool for such an analysis and various simulators have been developed by different agencies for freeway and/or arterial networks. For example, some well-known simulators for freeways include AIMSUN, FREFLO, FRESIM, FREQ, INTEGRATION, and KRONOS. While there have been individual tests of each software by various agencies, no comprehensive effort has been made to quantify and evaluate the performance of each model. The major difficulty includes the lack of detailed real data and the time consuming manual effort to prepare input information for each software. To be sure, there is no user-interface developed to date that can generate input files for different simulators with a common set of data. Developing an integrated traffic analysis environment, where data processing, simulation, and output analysis can be automated as efficiently as possible, is of critical importance in improving traffic management and operations.

## 1.2 Research Objectives

The ultimate goal of this research is to develop a traffic simulation laboratory where various roadway design/operational alternatives can be evaluated with state-of-the-art traffic simulators under an integrated database-simulation environment. The current research, phase I, evaluates existing freeway simulation models with real traffic data collected from the Minneapolis/St. Paul freeway system. The major tasks for the current phase include,

- Development of standard test cases for evaluating freeway traffic models.
- Data collection and development of database with test cases.
- Identification of freeway simulation software.
- Evaluation of selected software with test cases.

- Development of basic structure for automated freeway simulation environment.
- Maintenance and support for the freeway simulation software developed by the University of Minnesota.

### **1.3 Report Organization**

Chapter 2 reviews the major features of the selected simulation models, including hardware requirements, possible applications, modeling, input requirements, model parameters, and model outputs. The review results are summarized in the comparison at the end of the chapter. Chapter 3 explains the structure of the database developed to store the data for the various test cases. The criteria for model evaluation and a description of each test case are also included in Chapter 3. Chapter 4 contains the simulation results for each model for each test case. The comprehensive evaluation results for all the models based on the simulation results are also presented in Chapter 4. Chapter 5 develops a framework for the automatic simulation system including a prototype example. In Chapter 6, the improvements and support for the freeway simulator developed by the University of Minnesota are summarized. Finally, Chapter 7 includes the conclusions and future research needs.

## 2. OVERVIEW OF SELECTED SIMULATION MODELS

### 2.1 Selection of Simulation Models for Evaluation

In this research, six simulation models, currently available as computer software in the field, were selected for testing. Table 2.1 includes the selected simulation models and the software release versions used for testing.

MODEL	PROGRAM	FILE DATE	VERSION
Fresim	editor	11/4/94	5.0.0
	simulator	4/18/95	5.0.0
Aimsun2	editor	1/8/96	1.0.2
	simulator	1/11/96	1.0.2
Freflo	editor	3/3/92	1.0.0
	simulator	7/24/92	3.1.0
Freq11	editor	9/11/92	3.0.0
	simulator	11/2/92	3.0.0
Kronos	editor	6/21/95	8.2.0
	simulator	12/8/95	8.2.0
	display	6/13/96	9.0.0
Integration	simulator	10/12/95	2.0.6
	od matrix	8/30/95	1.2.c

Table 2.1 : Selected Simulation Models and Their Tested Software Release Versions

The following sections summarize the review results of the above models focusing on the following areas;

- hardware/software requirements.
- simulation modeling.
- model capabilities.
- model applications.
- input requirements.
- model parameters.
- output information.

## 2.2 FRESIM

### *Developer*

The FRESIM program was developed for the Federal Highway Administration as a part of the TRAF family of simulators.

### *Hardware/Software requirements*

The following are required for the FRESIM programs :

- An IBM compatible personal computer with a 386 processor.
- A math coprocessor.
- 5876 KB of memory.
- Up to 12 MB of disk space for programs and files.

### *Modeling overview*

FRESIM is a microscopic model which uses the PITT car following and gap acceptance logics developed at the University of Pittsburgh. As a vehicle enters the freeway, it is assigned a type, performance limits, driver type, and a desired freeflow speed. The freeflow speed is determined from an internal procedure based on which driver type was assigned to the vehicle. During simulation, each vehicle has an acceleration rate calculated based upon the position and speed of the target vehicle and the preceding vehicle. With the new acceleration rate a new vehicle speed is calculated then the target vehicle's position is updated using the calculated speed. This process is repeated for each time interval. Vehicle turnings are based on the percentages provided by the user.

Lane changes are based on the PITT gap acceptance logic. This consists of two parts. The first part is called the lead gap. This gap is based on three different deceleration rates :

- The deceleration rate to avoid colliding with the preceding vehicle.
- The deceleration rate three seconds after changing lanes.
- The deceleration rate of the preceding vehicle at maximum deceleration.



The second part is called the trailing gap. This is based on the risk of colliding with the following vehicle in the new lane. Vehicles will perform lane changes based on two conditions. The first condition is a mandatory lane change, which occurs when a vehicle has reached either the end of an acceleration lane, a lane drop, a lane blockage, or has passed the ramp exit sign for the vehicle's intended destination. With this type of lane change, the risk value increases as the need to change lanes increases. The second type is the discretionary lane change, which is to improve the driver's perceived situation and depends upon the driver type and the desired free flow speed.

### *Simulation capabilities*

The capabilities of the current version, as listed in the manual, are as follows;

- maximum of 19 time slices.
- up to 25 origin zones and 200 links.
- up to 475 origin/destination points.
- entry and exit ramp from either the left or right side.
- lane adds or drops on a link on the left or right side (3 max per link).
- radius of curvature.
- three types of auxiliary lanes.
- warning signs for lane drops, off ramps, and construction zones.
- three detector types for surveillance and incident detection.
- maximum of 20 incidents and construction zones.
- four types of ramp metering control options.
- user created vehicle specifications (16 max.).
- traffic flows in both directions.
- Multiple incident detection algorithms; California Algorithm, Payne's Algorithm 8, and double exponential smoothing

### *Input requirements*

The FRESIM program uses a series of data cards for data input. These data cards include :

- Initialization time period (sec.).
- Length of each time slice (sec.).
- Type and frequency of outputs.
- Freeway geometric data in a link/node form
- Position and type of auxiliary lanes per link.
- The gradient, superelevation and radius of curvature of the freeway.
- Queue discharge headway (sec.) and free-flow speed (mph) per link.
- Vehicle turning movements as a volume percentage for all exit ramps.
- Entry flows per time slice in vehicles per hour (vph) for the mainline entry point and all entry ramps.
- Incident placement and lane blockages.
- Incident capacity reduction factor.
- Time of incident.
- Lane add or lane drop information.
- Metering placement, metering type, and additional information.
- Traffic sensor type and placement.
- Incident detection specifications.
- Vehicle type specifications.
- Percent of heavy vehicles at freeway entry points.
- Pavement type of freeway.

### *Simulation parameters*

The following is a list of the model parameters used for model calibration :

- 10 car following sensitivity factors.
- Vehicle generation separation (sec.).
- Acceleration/deceleration lag times (sec.).
- Maximum deceleration rate (ft/sec<sup>2</sup>).
- Time to perform a lane change (sec.).
- Queue startup delays.

### *Output Information*

The FRESIM program has four categories of output:

- Input data tables.
- Intermediate statistics for every user specified time period.
- Cumulative statistics every time slice.
- Fuel and Pollution tables.

For each output category,

- A repeat of all card data.
- Fuel consumption rates (gal and mpg).
- Hydrocarbon, carbon monoxide, and nitrous oxide emissions (grams/mile).
- Number of vehicles entering and exiting a link.
- Number of lane changes and turning movements.
- Total distance traveled (veh-mi).
- Total travel time (veh-min) and delay time (sec/veh).
- Link volumes (vphpl), densities (vpmpl), and speeds (mph).
- A list of warning messages.

## 2.3 AIMSUN2

### *Developer*

Department of Statistics and Operational Research, Universitat Politecnica de Catalunya, Barcelona, Spain.

### *Hardware/Software requirements*

The following are required for the AIMSUN2 and GETRAM programs :

- An IBM compatible personal computer.
- A math coprocessor.
- 16 MB of memory.
- 4 to 6 MB of disk space for AIMSUN2 and GETRAM programs.
- Up to 5 MB of disk space for output files.
- Microsoft Windows NT.
- Exceed 5.0 or better.

### *Modeling overview*

The AIMSUN2 simulator is a microscopic simulator which is based upon the Gipps car following model. Vehicles arrive according to one of three random distributions :

- A negative exponential distribution.
- A shifted negative exponential distribution.
- A platoon distribution.

Vehicle are then assigned to a vehicle characteristic group based upon user specified information. Vehicles are not, however, assigned to a particular exit. Once on the network, the vehicle position is updated using a car following logic based upon the Gipps model. Two speeds are calculated for each vehicle for a distinct time period :

- Maximum speed a vehicle can accelerate to.
- Maximum speed a vehicle can reach based upon vehicle characteristics and preceding traffic.

The minimum of the above two values is selected as the new speed of the target vehicle and the vehicle position is updated according to that speed. This process is repeated for each vehicle for every time interval. Exiting traffic is based on the turning percentages prespecified by the user for every vehicle characteristic group.

Lane changes are based on the Gipps lane changing model. Lane changes have been divided into three different sections. The first section is the length of freeway between critical sections. Traffic conditions, such as speed and queue lengths, are used to determine whether a vehicle will change lanes. These are lane changes that a driver feels will improve his situation. The second and third zones are governed by the distance to an exit ramp or lane drop. Those vehicles within the second section will gravitate toward the required lane to exit the freeway or avoid a lane drop if possible. Within the third section, vehicles will change lanes to reach their destination even if they must slow down or stop to do so.

Traffic signals are modeled by adding a dummy vehicle at the stop line whenever the signal phase is red in order to continue using the car following logic. This same procedure is used to model the green time ramp metering option. The other ramp meter control, flow metering, is based on the maximum hourly flow rate.

### *Simulation capabilities*

The capabilities of the current version, as listed in the manual, are as follows;

- various vehicle classes, including HOV vehicles, based upon user specifications.
- various detector types.
- various traffic control devices.
- Origin/destination matrix information.
- left or right side entry and exit ramps.
- auxiliary lanes for ramps and lane add/drops.
- traffic in both directions.
- Output visualization of traffic propagation during simulation.

- incidents and construction zones.
- Graphical data entry program based on the Unix platform.

#### *Input requirements*

- The network geometrics.
- Entry flows rates(vph).
- Section speeds (kph).
- Section capacities.
- Vehicle characteristics.
- Turn movement percentages for each vehicle type.
- Ramp metering type and related information.
- Traffic control type and related information.

#### *Simulation parameters*

- Braking coefficients.
- Minimum distance between vehicles (m).
- Simulation step size (sec).
- Zone lengths for when a driver will consider a weave and when a driver must weave.
- Intersection visibility.
- Cruising speed tolerance.

#### *Output information*

The simulation results are given for both the entire network and/or for each section as specified by the user. The output include;

- Flow rates (vph).
- Densities (vpk).
- Speeds (vph).
- Travel times (hh-mm-ss).

- Delay times (hh-mm-ss).
- Stop times (hh-mm-ss).
- Number of stops per vehicle.
- Queue lengths.

Detector output depends upon the type of detector specified by user. These include :

- Vehicle counts.
- Occupancy rates.
- Speeds (kph).
- Densities (vpk).

## **2.4 FREFLO**

### *Developer*

The FREFLO program was developed for the Federal Highway Administration and is a part of the TRAF family of simulators.

### *Hardware/software requirements*

- An IBM compatible personal computer with a 386 processor.
- A math coprocessor.
- 4 MB of memory.
- Up to 8 MB of free disk space for programs and files.

### *Modeling overview*

FREFLO is a macroscopic simulator based on the high order continuum modeling approach. The momentum equation determines the section speed and consists of three parts. The first part is a convection term which estimates the tendency to remain at the current speed. The second part is a relaxation term which estimates the tendency to adjust

speeds to the equilibrium speed. The equilibrium speed is determined from one of three speed-density relationships. The third part is an anticipation term which estimates the tendency to change speeds depending on changes within the downstream traffic conditions. Finally, flow rates are determined by the product of the section speed and the section density.

#### *Simulation capabilities*

- maximum of 19 time slices.
- up to 256 links.
- traffic in both directions.
- right sided entry and exit ramps.
- up to 20 incidents or construction zones.
- HOV lanes.

#### *Input requirements*

- Freeway geometric data in node-link form.
- Link capacity and free-flow speed.
- Link speed-density relationship type and coefficients.
- Flow entry rates (vph) and turning percentages for exit ramps.
- Traffic composition, i.e., percentage of truck and carpool traffic.
- Length of time slice (sec).
- Jam density (vpmpl).
- Node coordinates (for graphics package only).
- Link curvature (for graphics package only).



### *Model parameters*

The following is a list of the model parameters used for calibration :

- The capacity (vph) of each segment.
- Speed-density relationship coefficients depending on the choice of function types, i.e., cubic polynomial, reciprocal cubic polynomial, or a three regime speed-density relationship.
- Relaxation time and anticipation coefficients for the momentum equation.

### *Output information*

The FREFLO program allows the user to select the type of output and time intervals that the output is needed. Intermediate output include :

- Vehicles discharged per link by vehicle classification.
- Volume (vph) for both regular use lanes and HOV/bus lanes.
- Speed (mph) for each vehicle and lane classification.
- Density (vpmpl) for each vehicle and lane classification.

Cumulative output include :

- Vehicle miles traveled.
- The number of vehicle trips.
- Person miles traveled.
- The number of person trips.
- Average volume (vph).
- Average speed (vph).
- Delay time per vehicle (sec/veh or min/veh).
- Total delay for vehicles (veh-min or veh-hr).
- Total delay for persons (person-min or person-hr).
- Travel time per vehicle (sec/veh or min/veh).
- Total travel time for vehicles (veh-min or veh-hr).
- Total travel time for persons (person-min or person-hr).

## 2.5 FREQ11

### *Developer*

Institute of Transportation Studies, University of California at Berkeley.

### *Hardware/software requirements*

- An IBM compatible personal computer with a 386 processor.
- A math coprocessor.
- 8 MB of memory.
- Up to 5 MB of free disk space for programs and files.

\* The FREQ11 program does not work with the Microsoft memory manager due to conflicts with FREQ11's memory manager. \*

### *Modeling overview*

The FREQ model assumes that, under freeflow conditions, all vehicles that enter a section will leave a section during the same time period. First, the model searches the freeway for weaving sections. Then the weaving zone lengths are calculated. Next the weaving capacities are determined and the capacity values of all the sections within the weaving zone are set to this capacity. Next the model searches the entire freeway for the sections in which the flow rate exceeds the capacity of each section. It should be noted that the model converts the specified flow data into a set of origin/destination matrix to determine the volumes of each section in the main freeway. The estimated O/D matrix is redistributed if any bottleneck sections are found. Finally, the speed of each section is determined from the speed-V/C curve and the mainline queues are determined. FREQ treats each time period separately with the queue data from the previous time period as the initial condition.

### *Simulation capabilities*

- maximum of 24 time slices.
- up to 158 sections.
- maximum of 78 on-ramps and 78 off-ramps.
- entry and exit ramps from either the left or right side.
- capacity variations per time slice for each segment to simulate incidents and construction zones.
- up to 9 lanes per section.
- System or local flow optimization ramp metering.

### *Input requirements*

- Number of time slices for simulation.
- Number of time slices per hour.
- Freeway and arterial street geometric data
- Flow rates per time slice in vehicles per hour (vph) or in vehicles per time slice.
- Gradient of the freeway.
- Free-flow speed and capacity for each section (mph).
- Speed vs. V/C ratio curve.
- Truck percentages for emission calculations only.
- The modal shift sensitivity rating (high, medium, low).
- The perceived travel time savings.
- Emission and fuel consumption rates (manual or EPA tables).
- The arterial signalization progression rate (good, poor, or no signal).

### *Model parameters*

The following is a list of the model parameters used for model calibration.

- The capacity (vph) of each segment.
- The V/C ratio for bottlenecks.
- The speed vs. V/C ratio curve.

### *Output information*

The FREQ11 program has many output tables and graphs available which the user can select from. These include :

- Contour maps of speed, density, queuing, V/C ratio, and six environmental factors.
- Origin/destination tables for travel times (min.) and flow rates.
- Freeway statistics for each time slice for each segment.
- Cumulative statistics for the freeway.
- Freeway geometrics table.

The freeway statistics table includes :

- Adjusted flow rates (vph).
- V/C ratios.
- Queue lengths.
- Storage rates.
- Weave effects (vph).
- Freeway travel times (veh-hr).
- Ramp and freeway delays (veh-hr).
- Total travel times (veh-hr).
- Total distance traveled (veh-mi).
- Fuel consumption (gal.).
- Hydrocarbon, carbon monoxide, and nitrous oxide emissions (kg).

## 2.6 KRONOS

### *Developer*

University of Minnesota, Minneapolis.

### *Hardware/Software requirements*

- An IBM compatible personal computer with a 386 processor.
- A math coprocessor.
- 8 MB of memory.
- Up to 5 MB of free disk space for programs and files.

### *Modeling overview*

KRONOS is a macroscopic simulator based on simple continuum modeling approach. It estimates the macroscopic traffic parameters, i.e., flow, speed and density, of every 100 ft segments in a given freeway every one second. The behavior of interrupted flow, such as merging, diverging and weaving, is modeled by defining internal boundaries and by explicitly estimating the amount of flows crossing the boundaries considering the current traffic conditions and capacity of the segments. Further, the right-most lane flows in some segments are treated separately from the mainline flow to reflect the drivers' behavior near exit ramps.

### *Simulation capabilities*

The following is a list of program capabilities :

- maximum of 200 time slices.
- graphical input/output module.
- up to 100 freeway segments or 120,000 feet of freeway.
- maximum of 8 through lanes per segment.
- maximum of 25 on-ramps and 25 off-ramps.
- multi-stage incidents

- left and right sided entry and exit ramps.

### *Input requirements*

The following is a list of input requirements :

- Freeway geometric data
- Freeway demand entering and exiting each ramp (vph) for each time slice.
- Segment capacities (vphpl).
- Initial traffic conditions for segments (optional)
- Speed-density /flow-density relationship

### *Model parameters*

The following is a list of the model parameters used for model calibration :

- The capacity (vph) of each segment.
- The flow-density relationship.

### *Output information*

The KRONOS output types consist of graphics, spreadsheets, and the tables for Measure of Effectiveness statistics;

- Two dimensional graphs of flow, density, and speed against distance.
- Three dimensional graphs of flow, density, and speed against distance and time.
- Emulation of density variation through time for a given freeway.
- Spreadsheet-format files containing flow, speed and density values for every 100 ft segment for each time interval specified by user.
- MOE tables including Total Travel (veh-miles), Total Travel Time (veh-hr), Average Speed (mph) and Delay of each segment (veh-hr).

## 2.7 INTEGRATION

### *Developer*

Queen's University located in Kingston, Ontario, Canada.

### *Hardware/software requirements*

The following are required for the INTEGRATION and QUEENSOD programs :

- An IBM compatible personal computer with at least a 386 processor.
- A math coprocessor.
- 16 MB of memory.
- Up to 12 MB of free disk space for programs and files.

### *Modeling overview*

The INTEGRATION simulator combines both microscopic and macroscopic approaches to model vehicle behavior in a freeway network. It employs its own car following, lane changing, and traffic assignment logics. Before simulation begins, every vehicle entering the network is assigned a vehicle number, vehicle class, an origin, a destination, and a departure time based on an origin/destination matrix. Each vehicle, as it enters the network, selects a lane based primarily on vehicle headways. Then, for every tenth of a second, the space headway between the target vehicle and the preceding vehicle is calculated based upon its own car following logic. Based on the estimated headway, the vehicle speed is determined from a macroscopic speed-headway relationship provided by user. Finally, using the new speed, the vehicle position is updated. At the end of a link, the vehicle makes a route decision based upon internally created origin/destination path trees. Traffic signals are simulated using a dummy vehicle to stop traffic. Whenever a signal is red, the dummy vehicle is placed at the end of the link. This result in a reduction in headways causing the true vehicles to stop and form queues. The origing/destination matrix can be created using the QUEENSOD software, which uses using either the least

square error (LSE) method or the least relative error (LRE) method to determine the optimal estimates of the origin/destination flows with a given set of real data.

### *Simulation capabilities*

The following is the summary of the current version capabilities found from the manual.

- five distinct vehicle classes including HOV vehicles.
- maximum of 32000 seconds (530 min.).
- freeway/arterial network with bi-directional flows.
- maximum of 3500 links.
- maximum of 350 origin and destination points (including ramps).
- maximum of 35 incidents per simulation.
- maximum of 350 traffic signals and/or ramp meters.
- maximum of 8 phases per control.
- maximum of 35,000 origin/destination pairs.
- maximum of 350,000 total vehicles.
- maximum of 7 lanes per link.

The following is a list of program capabilities for QUEENSOD:

- Perform the least square error model (LSE).
- Perform the least relative error model (LRE).
- Perform a maximum of 100 iterations.
- Handle up to 1000 links.
- Handle 5 path trees per time period.
- Handle up to 12 time periods.



### *Input requirements*

The INTEGRATION program does not have a data entry program unlike the other simulation packages. The DOS text editor is used to create the five required input files and a run control file. The following is a list of input requirements for the INTEGRATION program :

- Network geometric data as a series of links and nodes.
- Horizontal and longitudinal coordinates for each node within the network.
- Node identifier (origin, destination, intermediate).
- Upstream and downstream nodes for each link.
- Link capacities, jam densities, free-flow speeds, and speeds at capacity (speed-flow relationship data).
- HOV lane designation.
- Position of signals and ramp meters.
- Timing patterns for signals and ramp meters.
- Signal optimization data.
- Origin/destination matrix.
- Percentage of each vehicle type (up to 5 driver classes) including an HOV class.
- Time frame, position, and the number of effective lanes blocked for incidents.

Input requirements for the QUEENSOD software to estimate origin/destination matrices;

- Node file from the INTEGRATION program.
- Link file from the INTEGRATION program.
- Flow rate per link for each time period (vph).
- Free-flow travel time and observed travel time (sec.)
- Vehicle path trees for the network.
- Estimation method type (LSE or LRE).
- Number of iterations and dampening factor.

### *Model parameters*

The following is a list of the model parameters used for the calibration of INTEGRATION:

- The capacity (vph) of each link.
- The free-flow speed (kph) of each link.
- The speed at capacity of each link (kph).
- The jam density (vpk) of each link.

### *Output information*

The available output data include;

- On screen graphical animation of vehicle movements.
- Run error file.
- Labeled output files.
- Unlabelled output files.

The on screen animation occurs while the simulation is running. The entire network is drawn and boxes representing each vehicle are added as the vehicles enter the network. Each vehicle has one of the four colors representing the region of the speed-flow curve the vehicle is operating under. Also, the number of departures, arrivals, and vehicles traveling the network are given along with the elapsed simulation time. The run error file contains any errors that occurred during the simulation. Also included are the limitations of the program and the execution time.

The labeled and unlabeled output files contain simulation outputs. The labeled output file is for easier comprehension of the simulated results. The unlabeled output files can be used in future INTEGRATION runs. The output data included in those files are;

- Signal optimization plans.
- Origin/destination travel time tables (min.).
- Number of vehicles exiting each link.
- V/C ratio for each link.
- Travel times for each link (min.).

- Total network travel time (veh-hr).
- Total network distance traveled (veh-km).
- Average trip time per vehicle (min.).
- Average trip distance per vehicle (km).
- Total number of stops within the network.
- Average link speed (kph).
- Total number of vehicles that entered, exited, and were still en-route within the network.
- Minimum path trees (unlabelled only).
- Fuel consumption (liters).
- Hydrocarbon, carbon monoxide, and nitrous oxide emissions (g).
- Average occupancy per link.
- Probe vehicle report (travel time and pollution index).

The QUEENSOD program output files are also labeled or unlabeled. Again, the unlabeled files can be used in future INTEGRATION runs. Outputs for this program include :

- Error per iteration.
- Difference between observed and estimated flow rates for each time period.
- Estimated flows which exceed capacity.
- Link flows discarded due to flow discontinuity
- Origin/destination file for use with the INTEGRATION program.

## 2.8 Summary of Simulation Models

This chapter briefly reviewed the six simulation models selected in this research. The following tables summarize the major features of each model.

	FRESIM	AIMSUN 2	FREFLO	FREQ11	KRONOS	INTEGRATION
<b>SYSTEM NEEDS</b>						
operating system	ms-dos	Windows NT V3.5	ms-dos	ms-dos	ms-dos	ms-dos
minimum memory required	6 MB	16 MB	4 MB	8 MB	640 KB	16 MB
disk space required	12 MB	16 MB	8 MB	5 MB	5 MB	12 MB
required support programs		Exceed V5.0				
data entry program	text-based	graphical	text-based	partial graphical	graphical	none
origin/dest. matrix	optional	optional	no	no	no	required : a program that converts flows to o/d matrix is available
microscopic	yes	yes				
mesoscopic						yes
macroscopic			yes	yes	yes	

Table 2.2 : Simulation Model Computer Requirements and General Information of the Selected Models.

ITEM	FRESIM	AIMSUN 2	FREFLO	FREQ11	KRONOS	INTEGRATION
<b>SIGNS</b>						
stop/yield signs	no	yes	no	no	no	yes
guide signs	yes	no	no	no	no	no
variable message signs	no	yes	no	no	no	yes
<b>RAMP METERING</b>						
traffic responsive	yes	yes	no	no	no	no
fixed time	yes	yes	no	no	no	yes
fixed capacity	yes	yes	no	no	no	no
flow optimization	no	no	no	yes	no	no
<b>HOV LANES</b>						
fixed placement	no	no	no	yes	no	no
user specified placement	no	yes	yes	no	no	yes
lane barriers	yes	no	no	no	no	no
<b>LANE BLOCKAGES</b>						
as capacity reduction	no	no	no	yes	no	no
as closed lanes	yes	yes	yes	no	yes	yes
incident detection	yes	no	no	no	no	no
<b>TRANSIT NETWORKS</b>						
full transit	no	yes	no	no	no	yes
full freeway	yes	yes	yes	no	no	yes
partial freeway	yes	yes	yes	no	yes	yes
single freeway with arterial	no	yes	no	yes	no	yes

Table 2.3 : Simulation Model Capabilities

ITEM	FRESIM	AIMSUN2	FREFLO	FREQ11	KRONOS	INTEGRATION
<b>GEOMETRICS</b>						
max. number of through lanes	5	20	9	9	8	7
model right side ramps	yes	yes	yes	yes	yes	yes
model left side ramps	yes	yes	no	yes	yes	yes
model lane add/drop zones	yes	yes	no	no	yes	no
model auxilliary lanes	yes	yes	no	no	yes	no
freeway grades	yes	yes	no	no	no	no
curve variables	yes	no	no	no	no	no
<b>DEMAND DATA</b>						
mainline exit volumes	no	no	no	yes	optional	yes
exit ramp volumes	no	no	no	yes	yes	yes
exit ramp turn percentage	yes	yes	yes	no	no	no
label flow as congested	no	no	no	no	yes	no
origin/dest data	optional	optional	optional	no	no	prefered
add vehicle/driver classes	yes	yes	no	no	no	preset choices
<b>SIMULATION TIME</b>						
maximum simulation time	3166 min	5940 min	3166 min	1440 min	6000 min	533 min
maximum time slices	19	65+	19	24	200	50
internal initialization ability	yes	yes	yes	no	no	no

Table 2.4 : Simulation Model Inputs

ITEM	FRESIM	AIMSUN2	FREFLO	FREQ11	KRONOS	INTEGRATION	
<b>PARAMETERS</b>							
capacity	no	yes	yes	yes	yes	yes	
jam density	no	no	yes	no	yes	yes	
freeflow speed	yes	yes	yes	yes	yes	yes	
speed at capacity	no	no	no	no	no	yes	
flow/density curve	no	no	no	no	yes	yes	
speed/density curve	no	no	3 types	no	no	no	
speed/VC curve	no	no	no	yes	no	no	
specialty parameters for the microscopic models	10 car following factors	braking coefficient					<p>note on INTEGRATION</p> <p>this model converts flow/den. relation to speed/headway relation.</p>
	queue startup delays	minimum distance between vehicles					
	maximum decel. rate	length of zone for mandatory lane changes					
	accel. and decel. lag times	length of zone for desired lane changes					
	minimum separation for vehicles	speed acceptance coefficient					<p>note on AIMSUN2</p> <p>included within the vehicle modalities are the vehicle length, maximum acceleration, maximum deceleration, mean desired speed, and the vehicle width.</p>
time to perform lane change	vehicle modalities						

2.5 : Simulation Model Parameters

ITEM	FRESIM	AIMSUN2	FREFLO	FREQ11	KRONOS	INTEGRATION
I-94 EB 3 hrs						
without graphics	2316	759	107	61	270	4377
with graphics		2189				5318
I-35W SB 36 min TEST 1						
without graphics	308	128	52	22	67	806
with graphics		409				965
HYPOTHETIC LANE DROP FWY 36 min TEST 2						
without graphics	140	112	34	12	39	427
with graphics		349				529

Table 2.6 : Simulation Run Times in seconds for Selected Test Cases Using a Pentium Personal Computer Running at 90 Mhz.

ITEM	FRESIM	AIMSUN 2	FREFLO	FREQ11	KRONOS	INTEGRATION
OUTPUTS						
speed	yes	yes	yes	yes	yes	yes
density	yes	yes	no	no	yes	no
flow rates	yes	yes	yes	yes	yes	yes
occupancy rates	no	no	no	no	no	yes
travel times	yes	yes	yes	yes	yes	yes
delay times	yes	yes	yes	yes	yes	no
distance traveled	yes	no	yes	yes	yes	no
queue lengths	no	yes	no	yes	no	yes
VC ratios	no	no	no	yes	no	yes
pollution index	no	no	no	yes	no	yes
GRAPHICS						
flow propagation	no	yes	need GCOR program	no	as density change	yes
2D speed, density, and flow graphics	no	no	no	yes	yes	no
2D queue lengths	no	no	no	yes	no	no
3D speed, density, and flow graphics	no	no	no	no	yes	no

Table 2.7 : Simulation Model Outputs.



### **3. DEVELOPMENT OF TEST CASES AND DATABASE**

This chapter develops a set of test cases with real data to evaluate the freeway simulation models selected in the previous chapter. The geometric and traffic data for each test case is stored in a database containing data files. The following sections describe the test cases developed with the real data collected from the Twin Cities freeway network and the structure of the database that store the data.

#### **3.1 Development of the Database Structure for Test Cases**

##### **3.1.1 Geometric Data Files**

The freeway geometric information were extracted from the photo data logs developed by the Minnesota Department of Transportation. Unfortunately, these photo data logs do not show where the detectors are located. Thus, the location of each detector was determined from the detector placement maps provided by the Minnesota Department of Transportation. All the geometric data for each test section was stored as the Microsoft Powerpoint files, which show the detailed geometric information for each case in graphical format. Each file includes detailed length and detector information such as their position and their corresponding detector station.

##### **3.1.2 Traffic Data Files**

The traffic data files containing the loop data, i.e., 5-minute volume and occupancy for each detector, were developed using the Microsoft Excel software and stored in the spread-sheet format. First, a set of ASCII data file including only the data from the detectors on the test sections were created using the data extraction software, which extracts the data from the selected detectors from the raw data files provided by the Traffic Management Center. The data extraction software, developed in the Department of Computer Science, University of Minnesota, converts the binary raw data file into an ASCII format for the selected portion of the data. The ASCII files were then imported into the Excel spread-sheet files, where the volume and occupancy data for each individual

detector are stored in a matrix format for each 5-minute interval. Each spreadsheet file contains 5-minute volume and occupancy data for each detector and station for a three hour period.

### **3.2 Development of Quantitative Test Cases**

#### **3.2.1 Overview of Quantitative Testing**

In this research, the test cases were developed using real data collected from the field. Specifically, the volume and occupancy data collected on April 11, 1995, from the selected freeway sections were used to develop the test cases. The road surface was dry and the temperature was 32 degrees Fahrenheit. Detector numbers and locations were taken from detector placement maps provided by the Minnesota Department of Transportation. A three hour time period from 7:00 to 10:00 am was selected as this time included the morning peak-period. Each model was calibrated manually using the trial-and-error method with the real data and the difference between simulated and actual measured data from the internal check points on each test case were evaluated.

In order to get initial capacity estimates for the selected freeway sections, scatterplots were created plotting volume against occupancy. From these scatterplots, the shape of the flow-density curve could be estimated. Based on the data collected, the maximum flow usually occurred at the occupancy levels between 23 and 27 percent. Initial flow conditions were set equal to those conditions that existed for the first five minute time slice. Downstream demands, when called for by some simulation models, were set to the information of the closest detector station or combination of detector stations if ramps were involved.

#### **3.2.2 Evaluation Criteria for Quantitative Testing**

For the quantitative test cases, the following four error indices were calculated;

- The residuals, i.e., the difference between measured and estimated values for each time slice for each selected detector station.
- The mean absolute error of the residuals for each selected detector station.

- The error percentage for each time slice for each selected detector station.
- The mean percentage error for each selected detector station.

Formulation for these error indices are as follows :

$$\text{Residuals} = \text{measured} - \text{estimated}$$

$$\text{Mean Absolute Error} = \frac{\sum |\text{measured} - \text{estimated}|}{N}$$

$$\text{Error Percentage} = \frac{100 * (\text{measured} - \text{estimated})}{\text{measured}}$$

$$\text{Mean Percent Error} = \frac{\sum 100 * \left( \frac{|\text{measured} - \text{estimated}|}{\text{measured}} \right)}{N}$$

where,

N: the number of data points

### 3.2.3 Case 1 : I-494 Westbound South of Twin Cities International Airport

The first quantitative test is a 4.8 mile section of I-494 westbound stretching from the Minnesota River Bridge to Lyndale Avenue. This section is located just south of the Twin Cities International Airport and between the two largest shopping malls within the Twin Cities area; the Mall of America and Southdale. Figure 3.1 shows the geometrics of this test case. In this test case, all ramps are located on the right hand side of the freeway section. Further, despite all of the lane additions and drops, there are always three through lanes for the mainline traffic. At one point in the freeway, a 450 foot weave section is within 800 feet of an entry ramp with an acceleration lane. At another point, a 890 foot weave section is followed 800 feet later by a lane drop exit ramp. Both of these points are located at the Highway 77 cloverleaf. Flow rates along the freeway peaked between 7:30 and 8:05 am. Some sections do last longer, usually another 10 or 15 minutes. The flow rates steadily fall over the remainder of the time period except for a few short spikes around 9:00 am. The mainline section from Lyndale Avenue to the end of the

test section operated under congested conditions from 7:30 to 8:30 am. None of the ramps operated at a congested rate. Finally, an incident occurs near the Lyndale Avenue entrance ramp. This was a two car accident that blocked one traffic lane. The incident lasted about 10 minutes and started at 8:50 am. A more detailed geometric data set, the collected sensor data, the incident data, and the calibration parameters for each model can be found in Appendix A.

#### **3.2.4 Case 2 : I-494 Westbound Through Southwest Minneapolis Suburbs**

The second quantitative test is a 13.7 mile section of I-494 westbound stretching from Penn Avenue to I-394. This freeway section travels through the cities of Bloomington, Eden Prairie, and Minnetonka. This section has the longest length among the three test cases developed in this study. Also, no incidents occurred on this freeway section during the chosen three hour time period. See figure 3.2 for the geometrics of this test case. As with the first test case, all of the ramps are located on the right side of the freeway. Plus, only six different geometric types exist for this freeway. The freeway section begins with three lanes for the mainline traffic which is reduced to two lanes at the TH-100 cloverleaf. Also, there is a 400 foot weave section 800 feet downstream of this lane drop. This is part of the TH-100 cloverleaf. Past TH-100, the distance between successive ramps is at least 1000 feet with some being separated by over 7000 feet.

The traffic data collected from the test section indicate that the mainline flows for this case are uncongested throughout the three hour time period, although the flows near the beginning of the test section are heavy. Traffic flows peak between 7:30 and 8:30 am and gradually decline over the remainder of the time period. Some sections do show a minor increase in flow rates between 8:50 and 9:05 am. Ramp flows are operating, for the most part, within the uncongested portion of the flow-density relationship. But, some ramps near TH-100 and TH-169 do have some congestion periods. A more detailed geometric data set, the collected sensor data, and the calibration parameters used for each model can be found in Appendix A.

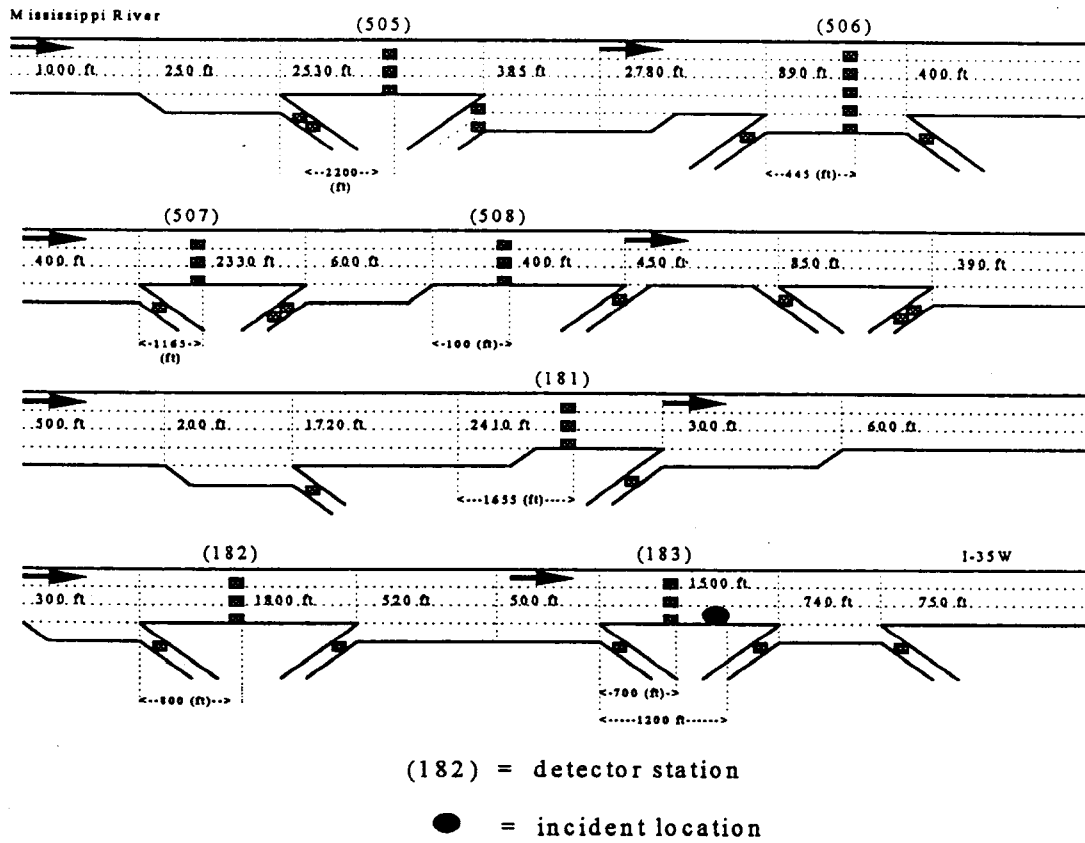
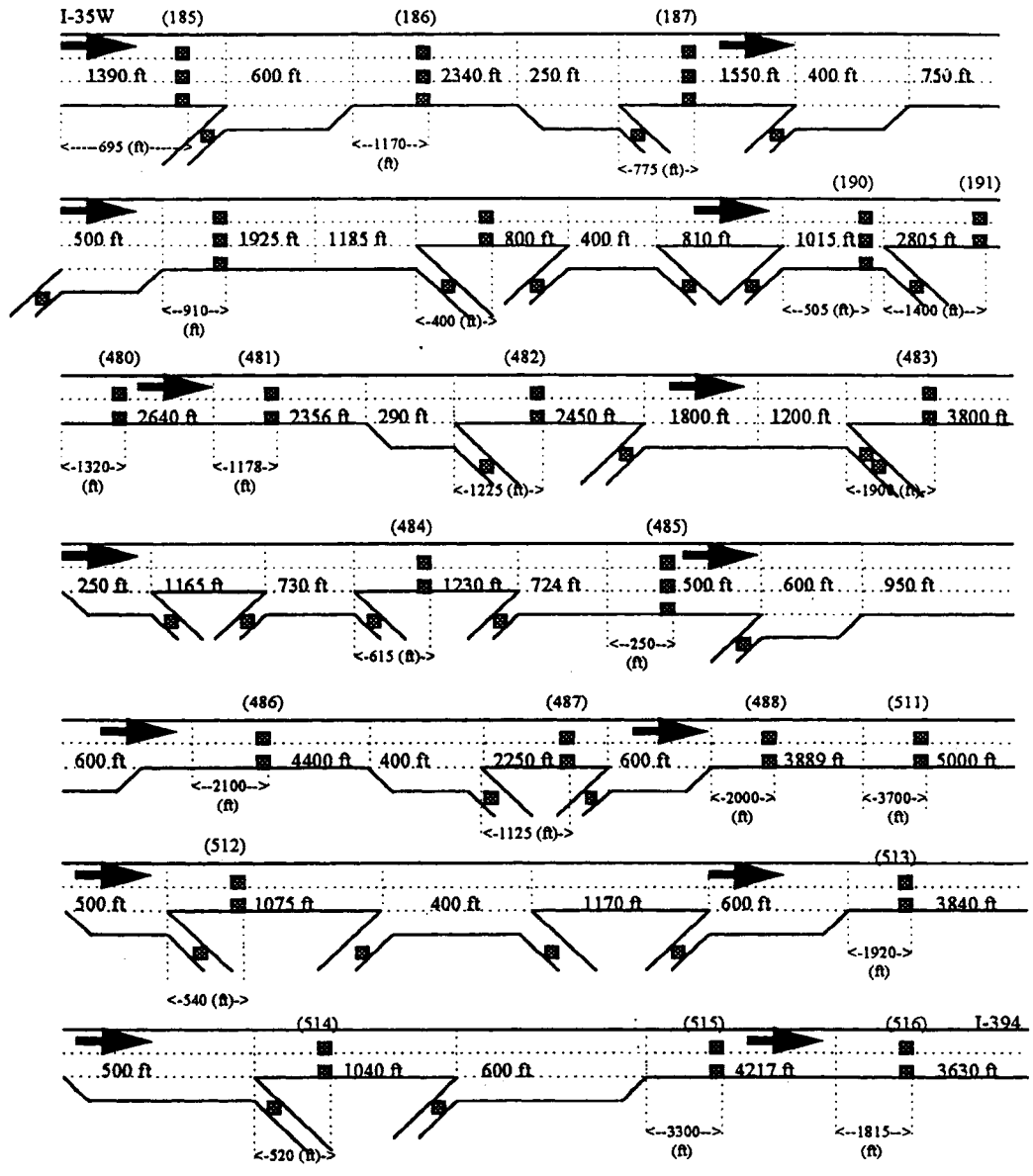


Figure 3.1 : Freeway Geometrics for Quantitative Test Case 1 ~ I-494 Westbound South of the Twin Cities International Airport



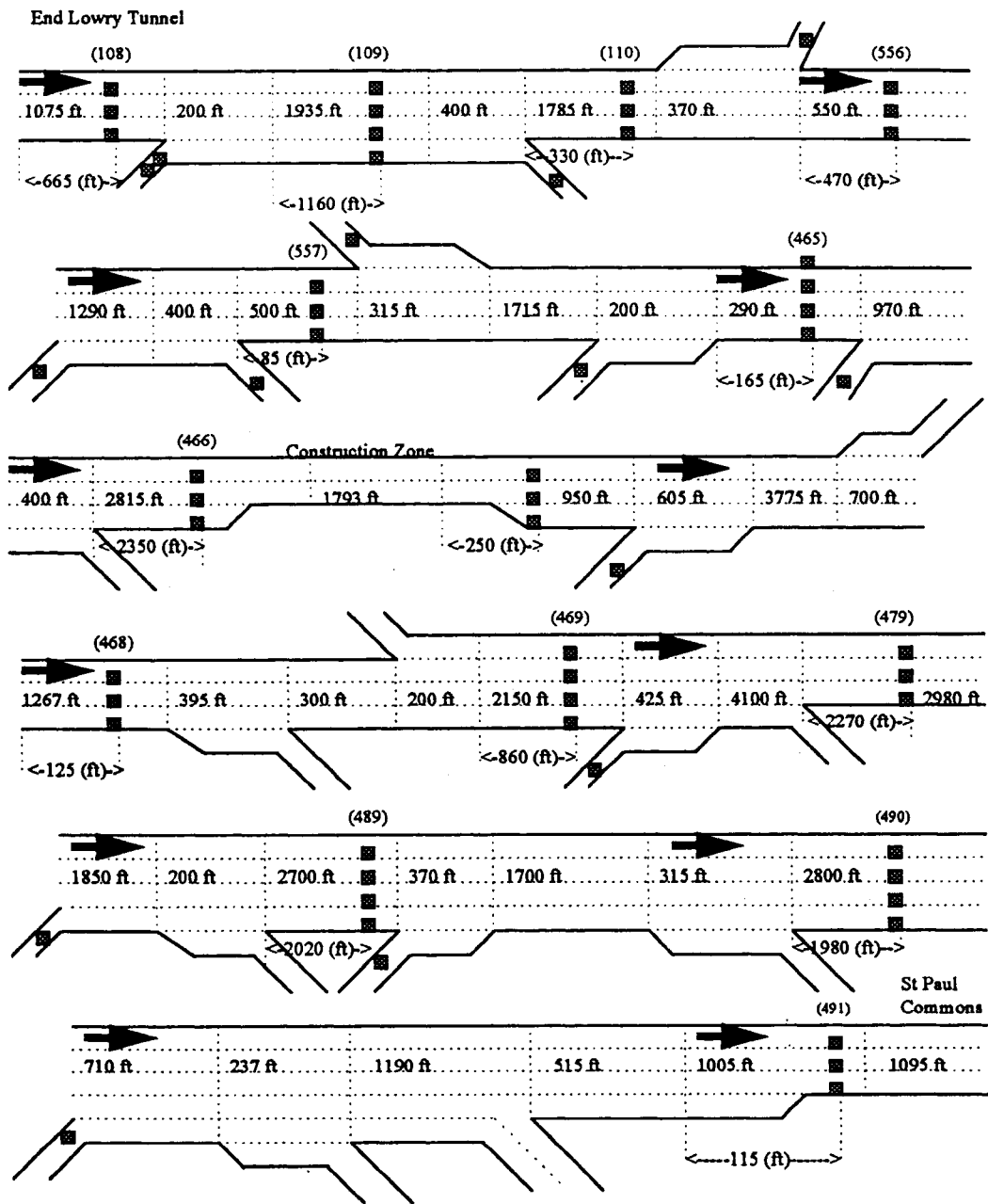
(514) = detector station

Figure 3.2 : Freeway Geometrics for Quantitative Test Case 2 : I-494 Westbound Through Southwest Minneapolis.

### **3.2.5 Case 3 : I-94 Eastbound Near University**

The last quantitative test is a 9.8 mile section of I-94 eastbound stretching from the west end of the Lowry Tunnel to John Ireland Boulevard. This section was selected as it includes a construction zone of approximately 0.8 miles that runs from 25th Avenue to Huron Street. No incidents were recorded during the time period. See figure 3.3 for the geometrics of this test case. Ramps are located on both sides of the freeway, although most are located on the right side. This freeway section starts out with three through lanes and increases to four lanes after the TH-280 entrance. Also, all of the left side ramps connect to either TH-280 or to I-35W. After TH-280, the geometrics have a steady pattern between entry and exit ramp distances as I-94 travels through St. Paul. On the Minneapolis side before the construction zone, there are three consecutive on ramps starting at I-35W. The next two entry ramps, at 6th Street and Cedar Avenue, are separated by 290 feet.

The construction zone reduces the number of mainline through lanes to two. This construction zone closed the entry and exit ramps between Riverside Street and Huron Street. This began during the first week of April, 1995. From the traffic data, it was discovered that the Huron Street entrance ramp was still open to traffic. All detectors within the construction zone were not operating. Traffic flows through the test case operated under uncongested conditions during the entire simulation time period. Traffic levels peaked between 7:45 and 8:15 am and dropped slightly before leveling off. In fact, some detector stations showed very little change between the peak period and the rest of the simulation time period. Also, the construction zone appeared to have little effect on the traffic flow based on upstream occupancy rates. Flows entering the construction zone were no higher than 75 percent of the total capacity. Ramp flows varied little for most ramps within the freeway section. The ramps near the beginning of the freeway did show some decline after 8:30 am. Otherwise, all ramps were operating under uncongested conditions throughout the simulation time period. The detailed geometric data set, the collected sensor data, and the calibration parameters used for each model can be found in Appendix A.



(489) = detector station

Figure 3.3 : Freeway Geometrics for Quantitative Test Case 3 ~ I-94 Eastbound During the University of Minnesota Ramp Reconstruction.



### **3.3 Development of Qualitative Test Cases**

#### **3.3.1 Overview of Qualitative Testing**

In this section a set of test cases were developed to evaluate the qualitative behavior of the simulation models, specifically focusing on the following areas;

- queue propagation and discharge behavior
- flow conservation.

A total of three cases were developed; two of them with the real geometrics and the other one with a hypothetical geometrics with multiple lane drop pipeline sections. For these tests, all simulation models were to use the same flow-density curve. The exception is the I-494 westbound test case for which calibration data was available. The common flow-density curve has a maximum flow of 2200 vphpl and operates at freeflow speeds until the density reaches 15 vpmpl. All test runs consisted of 18 two minute time slices for 36 total minutes of simulation.

#### **3.3.2 Evaluation Criteria for Qualitative Testing**

Two different types of tests were to be performed for the qualitative tests. These include shockwave and flow conservation tests. Shockwave tests were to test the ability of each model under different queue forming conditions. Flow conservation tests were to check if the models were operating correctly under uncongested conditions.

For the shockwave tests, the three test conditions were :

- A weaving section with heavy mainline and ramp flows.
- A dual lane drop situation.
- An incident blocking one complete lane under heavy flow conditions

Densities are collected every 200 feet for 18 two minute time intervals. Density tables are then created. From these tables, queue lengths and queue dispersion times are calculated. Queue lengths are considered to be the sum of all consecutive sections in which the estimated density is greater than one half the jam density. Dissipation measurements begin when the queue length reaches a maximum value and ends when the density at the beginning of the queue is reduced to an uncongested level.

The expected spillback lengths and dissipation times were determined from the following equations :

$$\text{number\_of\_vehicles\_in\_queue} = \frac{V_{cin} - \text{capacity}}{60} * T_{cong}$$

$$\text{spillback\_length} = \frac{\text{number\_of\_vehicles\_in\_the\_queue}}{\text{number\_of\_lanes} * K_{cong}}$$

$$\text{dissipation\_time} = \frac{\text{number\_of\_vehicles\_in\_the\_queue}}{\text{downstream\_dispersion\_rate}}$$

$$\text{downstream\_dispersion\_rate} = \frac{\text{capacity} - V_{uin}}{60}$$

Definitions of the terms in the equations are as follows :

- $V_{cin}$  : the entry flow rate during the congestion period.
- $T_{cong}$  : the time of the congestion period.
- $K_{cong}$  : the congestion density of a freeway section.
- capacity : the capacity of the bottleneck section.
- $V_{uin}$  : the entry flow rate after the congestion period.

To determine the spillback length, the number of vehicles each section could hold under the congested conditions was calculated and subtracted from the expected number of vehicles within the queue. Sections which could hold more vehicles than were available used the spillback length equation to determine the portion of the section which was congested. The total spillback length was the sum of all sections lengths in which congestion was present. The congestion density was calculated using the Greenshield's model.

For the flow conservation tests, flow rates are estimated every 200 feet for 18 two minute time intervals. Upstream and downstream points are to be selected based upon how far an unobstructed vehicle traveling at 30 miles per hour can go in two minutes. The difference between the two flow rates are expected to be close to zero.

### 3.3.3 Case 1 : I-494 Westbound at the TH-100 Cloverleaf

The first test using the I-494 test section was a shockwave test. Within the geometry of the section, there is a lane drop off-ramp followed by a type A weaving section. This weave section is within 800 feet of the dropped lane. This area is the TH-100 cloverleaf. See figure 3.4 for the geometrics of this test case.

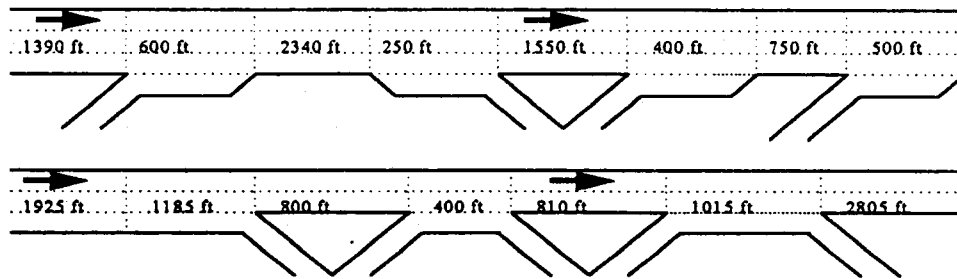


Figure 3.4 : Freeway Geometrics for Qualitative Test Case 1 ~ I-494 Westbound at the TH-100 Cloverleaf

As this section was part of a longer quantitative test case, the calibrated values for capacity and other parameters were available. Initial entry flows for the mainline were set to 4500 vph and dropped to 2000 vph after 20 minutes. All ramp flows were set to 300 vph for the duration of the simulation except for the lane drop exit ramp and the weave sections. The lane drop exit ramp was set to 600 vph for the entire simulation time. For the second weave section, also a type A weaving section, the entry ramp was set to 300 vph for the entire simulation while the exit ramp was set to 1200 vph. The entry and exit flows for the first weave section start at 300 vph, increase to 1200 vph for 16 minutes, then returned to 300 vph. Spillback is expected at the lane drop exit due to capacity restrictions. The length of the queue should be approximately 4835 feet and take about eight minutes to clear. At the weaving section, very little spillback, if any, is expected. No spillback should occur at the second weaving section.

The second test for this freeway section is a flow conservation test. Three sections were selected for testing. The selected locations are :

- 790 feet before the first entry ramp to 600 feet past the first exit ramp.
- 940 feet before the second entry ramp to 200 feet past the lane drop exit ramp.
- 600 feet before the first weave section to 2200 feet past the second weave section.

These three sections all represent different types of ramps situations. Entry flow rates start at 2200 vph, raise to 4400 vph 12 minutes later, then drop to 1360 vph for the final 12 minutes. Entry and exit ramp flows range from 300 to 1200 vph depending upon the ramp type and the ramp position. From all of this, entry flows are expected to equal the exiting flows between the two points. The exception to this is the first time period after the flow rates change. Appendix B includes a detailed geometric figure, traffic data and parameters used for both tests.

### 3.3.4 I-34W Southbound Through Burnsville

There are two lane drops areas in this case. Both reduce the number of through lanes from three to two. Between these lane drops are three different ramp types, one which is a lane add entry ramp. The total distance between the two lane drops is slightly greater than 7500 feet. See figure 3.5 for the geometrics for this test case.

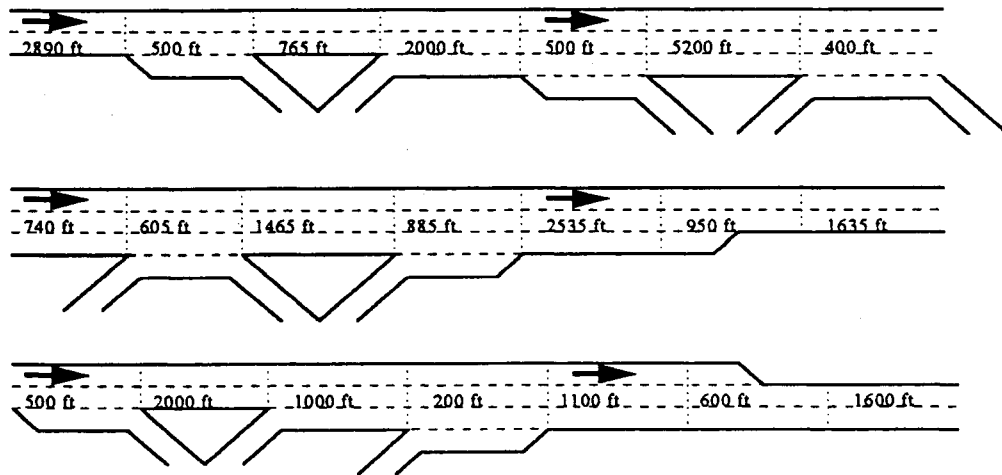


Figure 3.5 : Freeway Geometrics for Qualitative Test Case 2 ~ I-35W Southbound Through Burnsville

The first test for this freeway section is a shockwave test. Mainline flow rates start at 3800 vph and drops to 1360 vph after 20 minutes. All off ramps have a constant flow rate of 300 vph. On ramps start at either 300 or 600 vph and increase to 600 or 1200 vph, depending upon the location of the on ramp. Two queues are expected to form, one at each lane drop. The queue length at the first lane drop should be approximately 2620 feet long and require four minutes to clear. The second lane drop is expected to have a queue length near 3925 feet and require 9 minutes to clear. These queues are not expected to meet.

For the second test, an incident was placed 500 feet past the last lane drop. This was to block one traffic lane for 12 minutes. Flow conditions for all ramps were held constant for the entire simulation. These had values of either 300 or 600 vph. Mainline entry flows started at 3800 vph and then dropped to 1360 vph after the incident was cleared. The queue length is expected to be near 7915 feet with a dissipation time of 11 minutes.

### 3.3.5 Case 3 : Multiple Lane Drop Freeway Section

This test section is a pipeline section that begins with five lanes and has several lane drop sections in which the critical lane drops are 2000 feet apart. The minimum number of lanes is two. Also, no entry or exit ramps are included within this freeway section. See figure 3.6 for the geometrics for this test case.

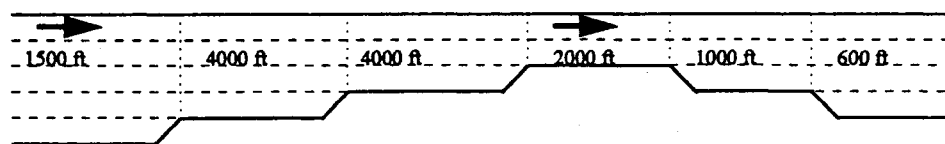


Figure 3.6 : Freeway Geometrics for Qualitative Test Case 3 ~ Multiple Lane Drop Freeway Section

The first test for the multiple lane drop section is a shockwave test. Entry flows start at 1100 vph and rise to 8250 vph for 30 minutes before dropping back to 1100 vph. Two

queues are expected to form. The first queue is expected where four lanes becomes three lanes. The second queue should form where three lanes become two. Also, the second queue should meet the first queue length. The combined length of both queues is expected to exceed 9500 feet and require 14 minutes to clear.

The last test is another incident test. Flow rates start at 4400 vph and drop to 1100 vph after the incident is cleared. The incident occurs at the midpoint of the two lane segment and completely blocks one lane for 12 minutes. A queue of 6200 feet is expected with a dissipation time of eight minutes.

### **3.4 Calibration of Model Parameters**

Calibration procedures are fairly similar for the macroscopic models. All three require the user modify various relationship curves and freeway section capacities. KRONOS and FREQ11 were the simplest to calibrate. These two require data points for their respective relationship curves. Then the segment capacities are changed. FREFLO requires the user to determine the coefficients for the selected speed/density equation and then change the link capacities. INTEGRATION takes a macroscopic approach in calibrating the microscopic logic used for simulation. This consists of entering three points of a speed/density curve which the model converts into a speed/headway curve. Then the link capacities are changed. FRESIM uses a series of car following sensitivity factors. These are a part of the car-following logic which calculates the vehicle acceleration. These are not easy to determine and many runs can be necessary. AIMSUN2 is affected by a number of items. These include the vehicle classes created in the TEDI program, the car-following parameters, the lane change zone distances, and to some extent, the segment capacities. Getting the right parameter combinations can take many runs.

#### **3.4.1 FRESIM**

The FRESIM model uses a number of different factors for calibration. These appear on three different data cards. The first card allows the user to change the car following sensitivity factors for the 10 internal driver types. These are the main parameters and

affect the calculation of the acceleration rate for a vehicle. The second card allows the user to change the pavement coefficients, the acceleration lag time, and the deceleration lag time. The last card sets the time to complete a lane change maneuver, the minimum separation time for the generation of vehicles, the percentage of drivers that will yield to a merging vehicle, and the maximum non-emergency deceleration rate.

For these tests, the car following parameters had the most effect upon the simulation. A series of consecutive numbers were inserted for these parameters until the error rate between the estimated flows and the collected flows was minimized. The minimum separation time for vehicle generation was set by determining the generation time required to add a capacity flow at the first freeway entry link. The other values were set to the default values. The values used for the FRESIM model parameters can be found in Appendix A and Appendix B.

#### **3.4.2 AIMSUN2**

AIMSUN2 uses a combination of parameters for calibrating the model for a given freeway section. For the car following model, the speed acceptance percentage and the braking coefficient are used. The speed acceptance percentage is used to determine the desired speed of a vehicle which in turn is used to determine the maximum speed a vehicle can accelerate to over a time period. The braking coefficient is used to determine the maximum deceleration rate of a vehicle which is then used to determine the speed a vehicle can reach with respect to the preceding vehicle. There are also two parameters for the lane changing logic. These are distances, in meters, from an exit ramp that a driver will consider lane changes. Within the first distance, the lane change is determined by traffic conditions. Within the second distance, the lane change is determined by the driver's need to get to the exit lane. Even though this is a microscopic model, the capacities can also be changed have a minor effect upon the simulation results.

For this testing, the car following and lane changing parameters were changed until the simulated results were within an acceptable error rate when compared to collected data. Values used for the tests can be found in Appendix A and Appendix B.

### **3.4.3 FREFLO**

For FREFLO, the calibration process begins by selecting one of three speed-density relationships. The first is a cubic polynomial with four coefficients. The second is a reciprocal cubic polynomial which also has four coefficients. The last is a three regime discontinuous relationship. This uses two speed, two densities, and one coefficient. The simulation model can not determine the coefficients so they must be determined using a statistical package. Once the coefficients are determined, link capacities can be changed until the simulated output is within a reasonable error percentage of the real time data. If this does not happen, the coefficients can be changed and the process repeated. The values used for the various tests can be found in Appendix A and Appendix B.

### **3.4.4 FREQ11**

Calibration of the FREQ11 starts with choosing a speed vs. V/C ratio curve. The model has default curves and there is an option to create a new curve using gathered data points. Then, section capacities are altered until the simulated output is within an acceptable error rate when compared to real time data. If this does not happen, the speed vs. V/C ratio curve can be edited and the process repeated until an acceptable error level is reached. The values used for the calibration of this model can be found in Appendix A and Appendix B.

### **3.4.5 KRONOS**

The calibration of KRONOS mainly consists of revising the flow-density relationship, which is the major parameter for simulation. The relationship is defined by inputting the values for the jam density, the capacity, the density at capacity, and the flow and density rate at the end of the linear portion of the curve. Segment capacities are then altered until the simulated output is within an acceptable tolerance level when checked with real time data. If the altered capacities do not produce acceptable results, the flow-density curve should be reconfigured and the process repeated. The values used for the different tests can be found in Appendix A and Appendix B.



### 3.4.6 INTEGRATION

The parameters for the speed-headway curve are the main parameters to be calibrated in INTEGRATION for each link of the freeway. Parameters include the freeflow speed, the speed at capacity, the jam density, and the capacity. The freeflow speed is typically set to the speed limit. The other three are changed until the simulated output is within an acceptable error percentage of the real time data. Values used for the tests can be found within Appendix A and Appendix B.

### 3.5 Input File Preparation

Using the calibrated parameter values, a set of the input files were developed for each test case for each simulator. Table 3.1 shows the number of workdays required to complete the data preparation for the three quantitative test cases by a graduate student in the Department of Civil Engineering, University of Minnesota.

ITEM	FRESIM	AIMSUN2	FREFLO	FREQ11	KRONOS	INTEGRATION
geometrics	1	1	1	1/2	1/2	1
flow data entry	1-2	5-10	1-2	1	1	10-20
calibration	3-5	3-5	1-2	1-2	1-2	5-10

Table 3.1 : Data Entry Times in Workdays for the Quantitative Test Cases

It should be noted that the time needed for preparation of input files will vary depending on the size of the freeway network, the number of entry and exit points, the number of time periods, and the data process knowledge of an individual.

Geometric data entry for the six simulation models did not require much time to complete. In fact, no test case took longer than one workday to complete the geometric data input. Overall, those simulation models with graphical interfaces were found to be much easier to use and had faster data entry times.

Flow data entry was where the models had the largest difference based on data entry times. KRONOS and FREQ11 use actual flow rates for all entry and exit points and were found to be the fastest and easiest to use. FRESIM was somewhat more difficult to use. This was due to having to determine the flowrates downstream of exit ramps when detector data was not available for the section just downstream of an exit ramp. FRESIM can calculate the turning percentages internally when given the exiting volume and the downstream volume which does save some time. Adding entry flow data was just as simple with FRESIM as it was for KRONOS or FREQ11. FREFLO and AIMSUN2 require that data be entered for each time slice individually. For FREFLO, the user needs to leave the editing screen in order to get to the menu in which the time period can be changed. He or she will also need to reenter the entry and exit links for every time period. AIMSUN2 requires each time period to be a separate file. Here the user must scroll through the freeway and select the sections to edit. Finally, both require that the turning percentages for each exit ramp be calculated based on exiting flows and downstream flows. INTEGRATION needs to have the origin/destination matrix for a freeway. Each origin/destination pair must be added for each time slice. For large simulations, this can number in the thousands. If flow data is the only data available, a data file can be created from which the QUEENSOD program will estimate the origin/destination matrix. This file contains link flows and travel times for all links within the system and must be repeated for each time slice. A path tree file is required by the QUEENSOD program.

## 4. EVALUATION OF FREEWAY SIMULATION MODELS

This chapter performs the testing of the selected simulation models with the test cases developed in the previous chapter. First, quantitative testing was conducted with the test cases developed with the real data from the Twin Cities freeway network. Using each simulator, the traffic conditions of each test case were recreated using the time-variant external boundary conditions as the input information. The simulation results at the internal check points were compared with the data collected from the loop detectors in the test section. The qualitative testing was designed to evaluate the performance of each model in treating the propagation of congestion and in conserving traffic flows during the simulation. For the microscopic simulation models using stochastic distributions to determine the actual number of vehicles entering freeways, a total of six runs were made for each test case for each model to get the average simulation results. The following sections summarize the test results.

### 4.1 Quantitative Test Cases

#### 4.1.1 I-494 Westbound Short Section

Table 4.1 includes the mean percentage difference over a three-hour period for 5-minute volume between simulated and measured data with the first case, which is the I-494 westbound freeway section that passes the Twin Cities International Airport. Included are the mean average differences for the mainline detector stations and the overall average difference for each simulation model tested. Figures showing the comparison between the simulated and measured flow data at detector station 183 are also included in this section.

STA.	FRESIM	AIMSUN2	FREFLO	FREQ11	KRONOS	INTEGRATION
505	1.47	2.64	1.27	0.76	1.05	4.32
508	4.74	7.51	4.78	6.29	6.55	6.71
182	6.63	4.91	4.43	2.26	4.83	8.05
183	8.99	4.76	6.16	2.85	4.33	9.10
AVG.	4.71	4.46	4.41	3.04	4.44	7.30

Table 4.1 : Mean Percentage Difference for I-494wb South of Twin Cities International Airport

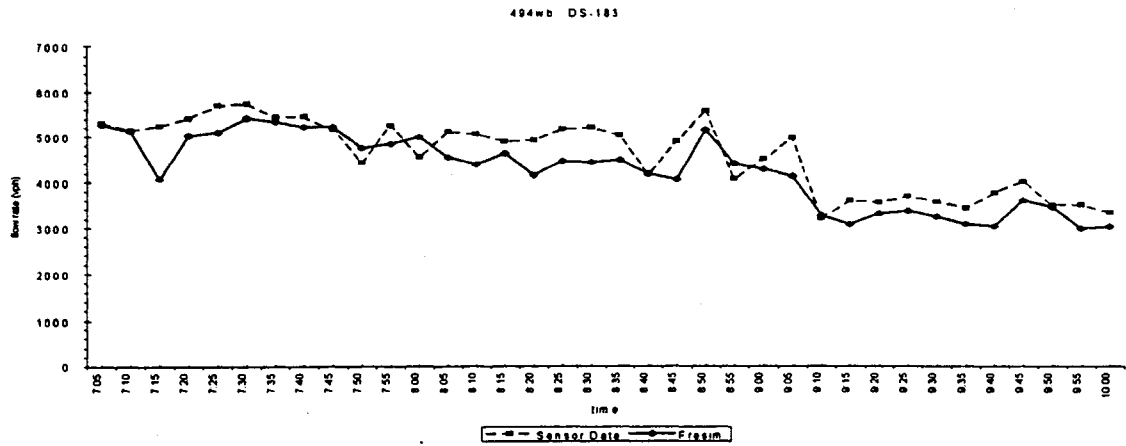


Figure 4.1 : Fresim Results Compared to Collected Flow Data at DS 183

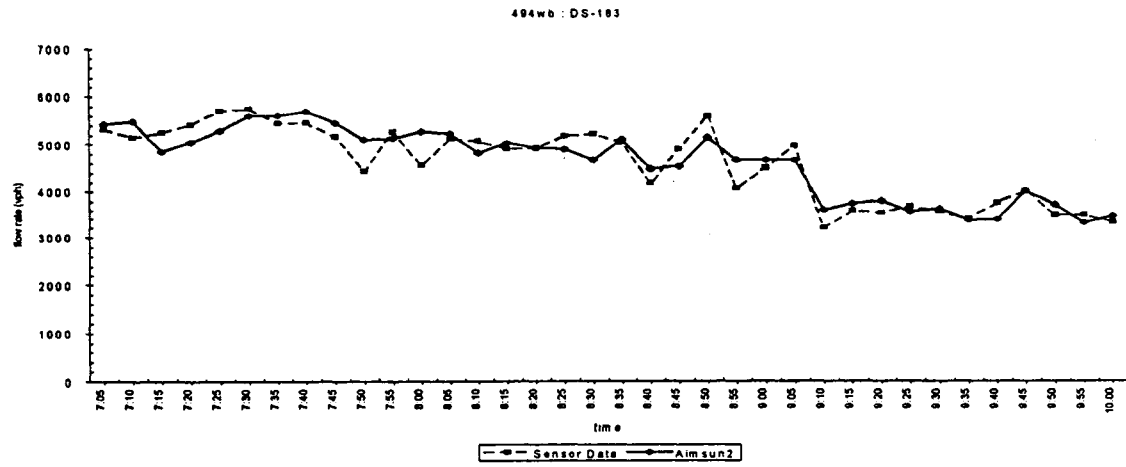


Figure 4.2 : Aimsun2 Results Compared to Collected Flow Data at DS 183

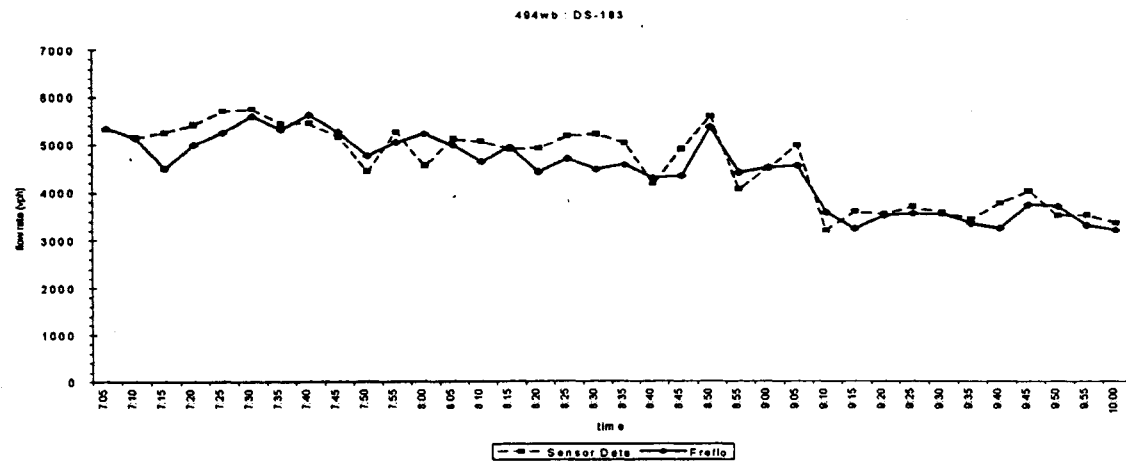


Figure 4.3 : Freflo Results Compared to Collected Flow Data at DS 183

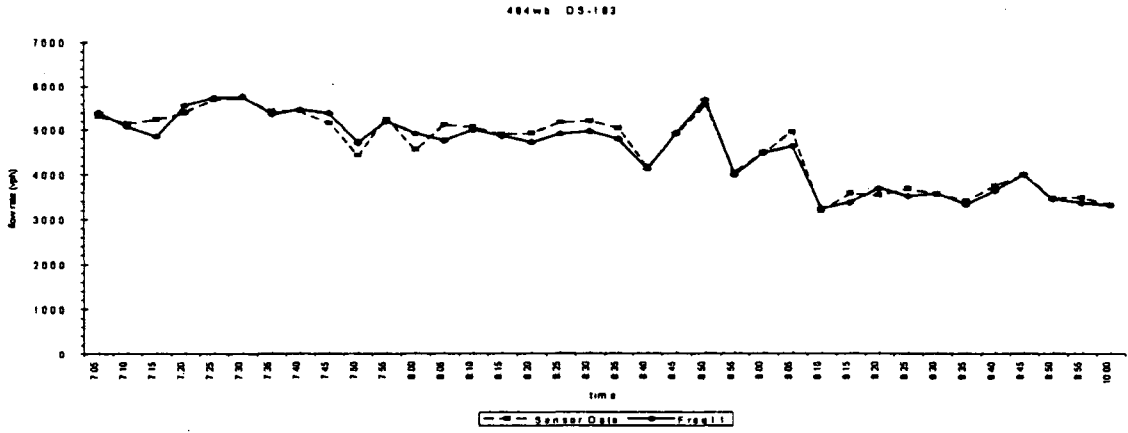


Figure 4.4 : Freq11 Results Compared to Collected Flow Data at DS 183

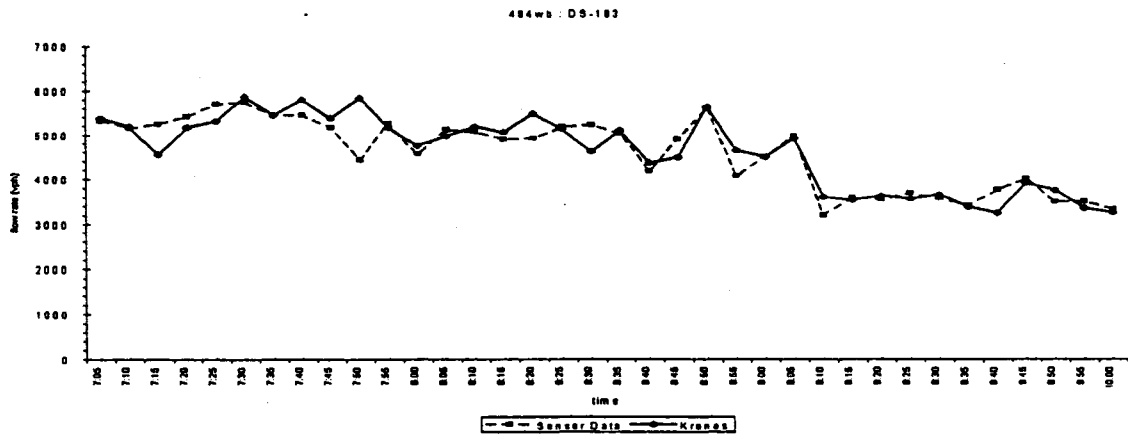


Figure 4.5 : Kronos Results Compared to Collected Flow Data at DS 183

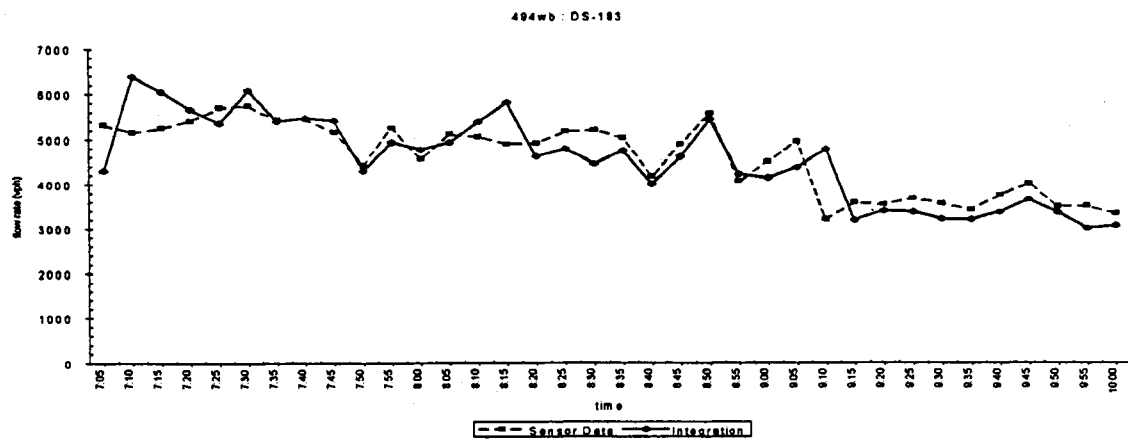


Figure 4.6 : Integration Results Compared to Collected Flow Data at DS 183

#### 4.1.2 I-494 Westbound Through the Southwest Minneapolis Suburbs

The following table shows the mean average difference for flow rates over a three hour period using five minute intervals for the I-494 westbound freeway section that passes through the southwest suburbs of Minneapolis. Included are the mean average difference for a series of mainline detector stations and the overall average difference for each simulation model tested. The figures showing the comparison results between the simulated and measured flow data at the last detector station, 515, are also presented in this section.

STA.	FRESIM	AIMSUN2	FREFLO	FREQ11	KRONOS	INTEGRATION
186	2.16	6.30	2.49	2.32	3.43	14.78
188	3.14	7.53	3.28	3.98	3.48	13.89
480	4.97	8.43	4.32	6.31	6.21	10.43
483	7.85	7.37	4.10	8.17	11.11	19.50
486	9.55	24.70	6.02	7.58	12.27	16.68
511	12.75	18.56	9.44	6.52	10.43	17.74
515	12.31	14.46	9.25	4.68	12.42	9.69
AVG.	7.53	12.48	4.56	4.65	8.48	14.67

Table 4.2 : Mean Percent Difference for I-494wb Through Southwest Minneapolis

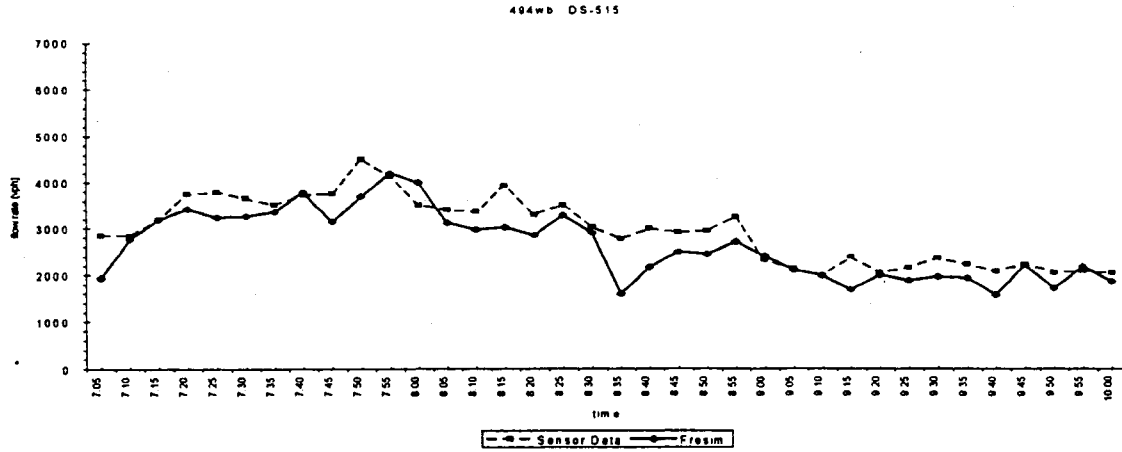


Figure 4.7 : Fresim Results Compared to Collected Flow Data at DS 515

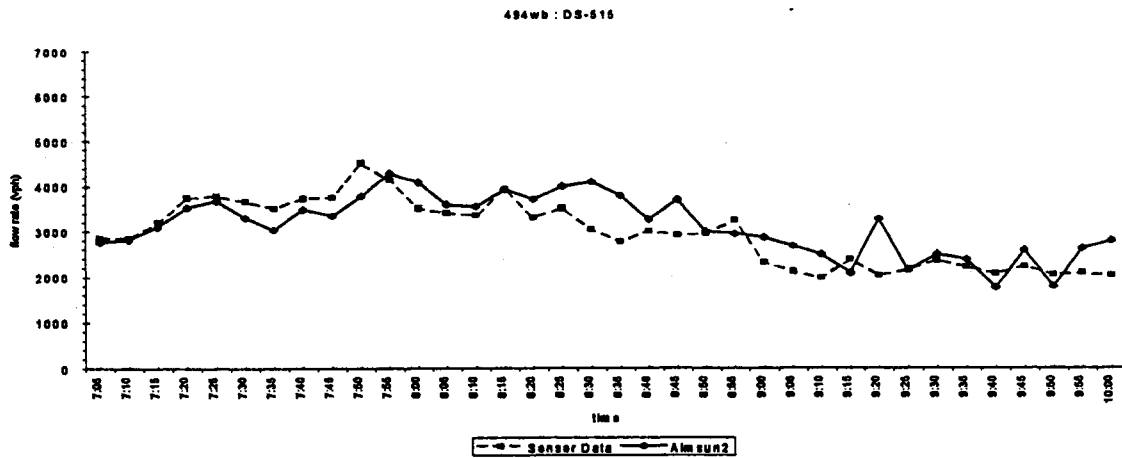


Figure 4.8 : Aimsun2 Results Compared to Collected Flow Data at DS 515

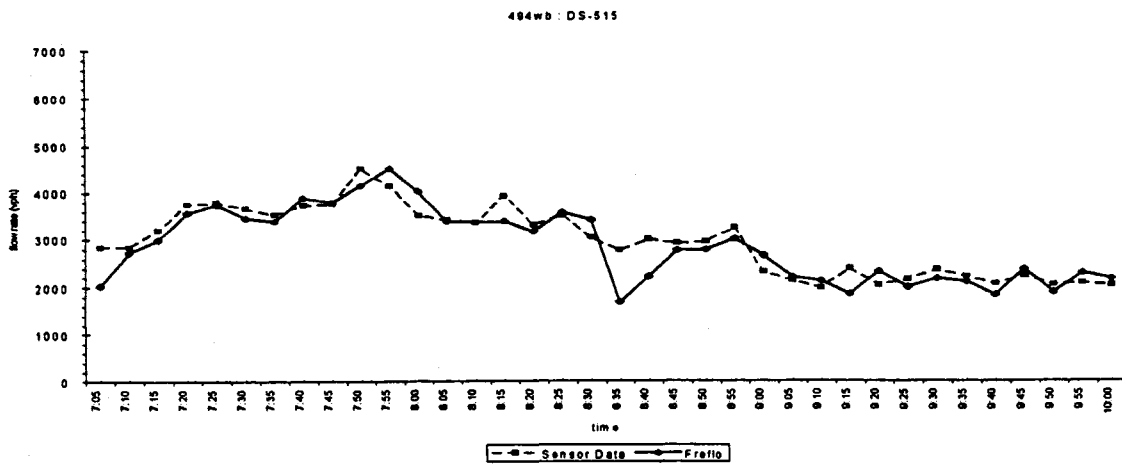


Figure 4.9 : Freflo Results Compared to Collected Flow Data at DS 515

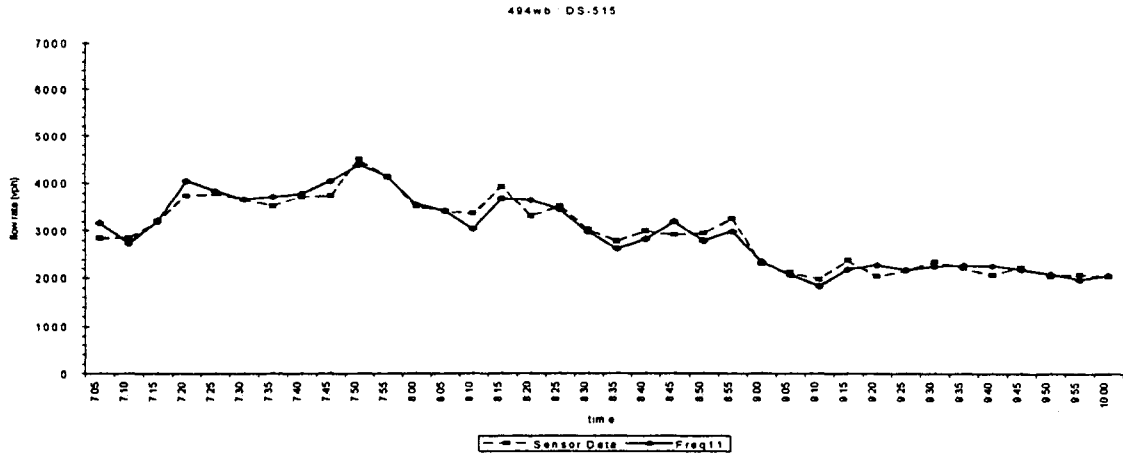


Figure 4.10 : Freq11 Results Compared to Collected Flow Data at DS 515

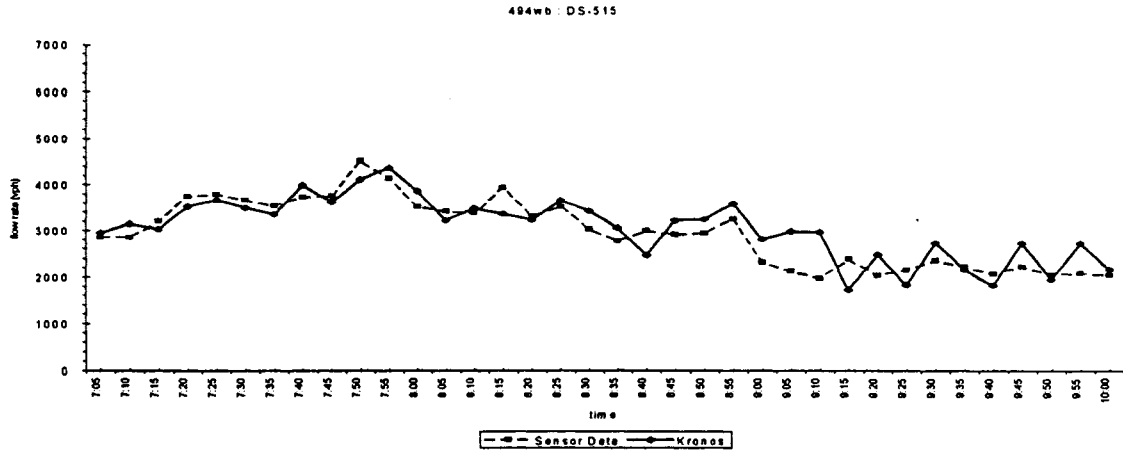


Figure 4.11 : Kronos Results Compared to Collected Flow Data at DS 515

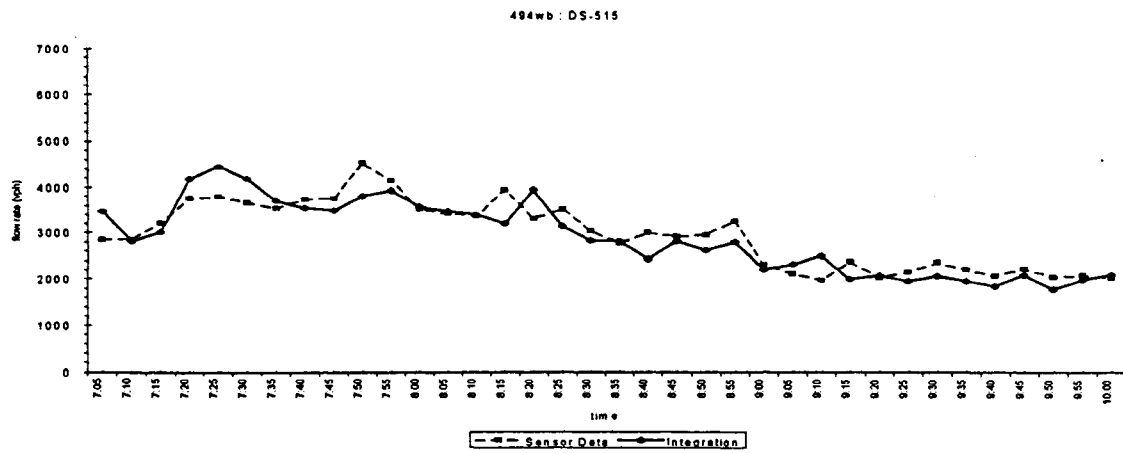


Figure 4.12 : Integration Results Compared to Collected Flow Data at DS 515



### 4.1.3 I-94 Eastbound During University of Minnesota Ramp Reconstruction

The following table shows the mean percentage differences between the simulated and measured 5-minute flow rates over a three hour period for the I-94 eastbound section which includes a construction zone near the University of Minnesota. The values in the table are the mean average differences at the mainline detector stations and the overall average difference for each simulation model tested. The figures in this section show the difference between measured and simulated flow rates at the detector station 467, which is located at the downstream of the construction zone.

STA.	FRESIM	AIMSUN2	FREFLO	FREQ11	KRONOS	INTEGRATION
110	3.65	7.16	2.79	4.48	3.97	7.13
557	6.49	8.45	4.27	4.92	8.57	7.94
467	6.25	10.11	4.56	6.13	6.83	8.54
468	6.65	12.99	4.39	6.38	9.39	8.66
479	8.63	11.53	6.13	4.28	9.58	9.24
490	10.12	10.72	6.95	4.75	8.53	9.91
491	11.94	16.48	7.81	4.48	13.87	7.21
AVG.	7.68	11.06	4.56	4.49	8.68	8.38

Table 4.3 : Mean Percentage Difference for I-94eb Through the University of Minnesota Ramp Reconstruction

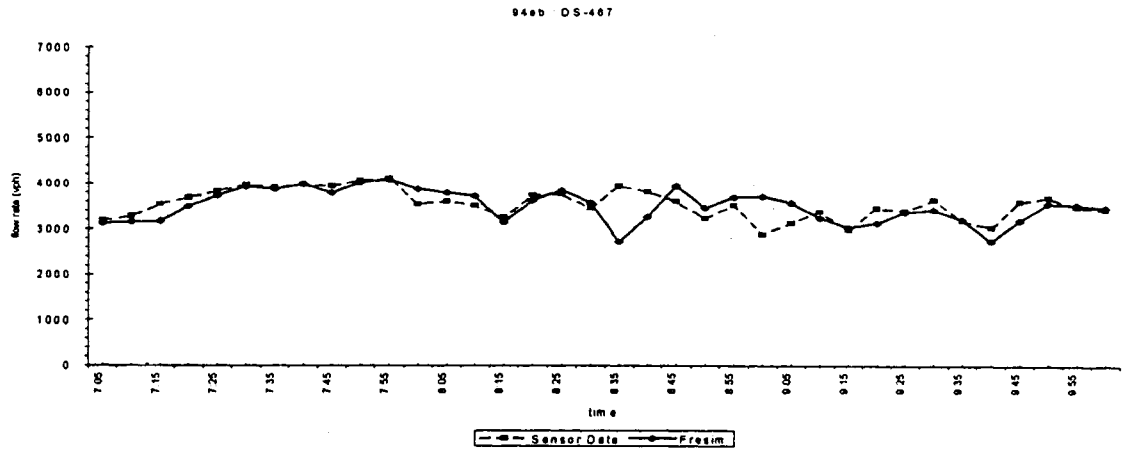


Figure 4.13 : Fresim Results Compared to Collected Flow Data at DS 467

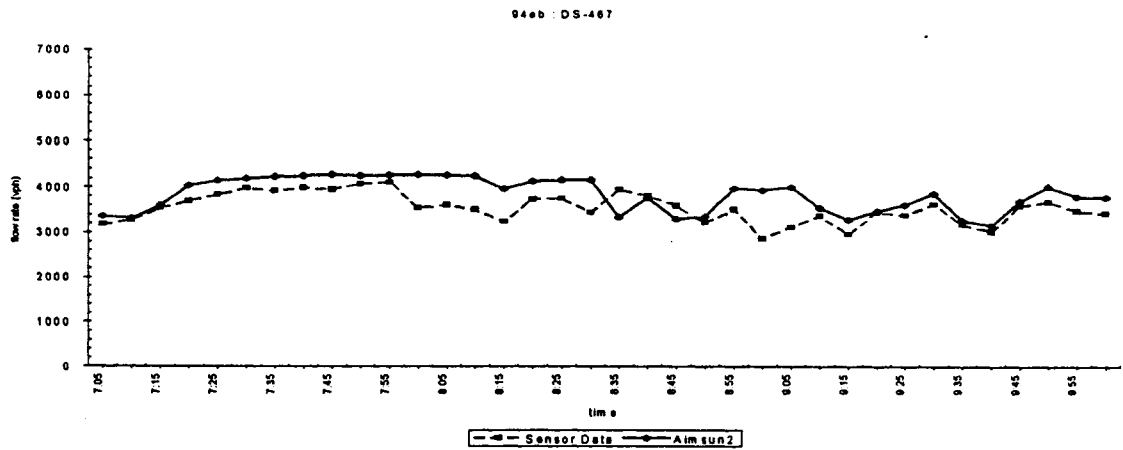


Figure 4.14 : Aimsun2 Results Compared to Collected Flow Data at DS 467

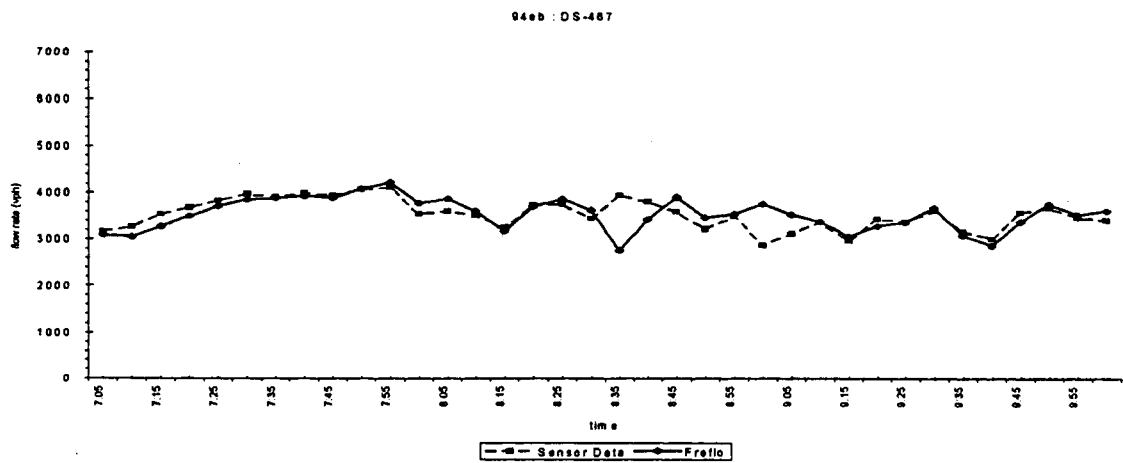


Figure 4.15 : Freflo Results Compared to Collected Flow Data at DS 467

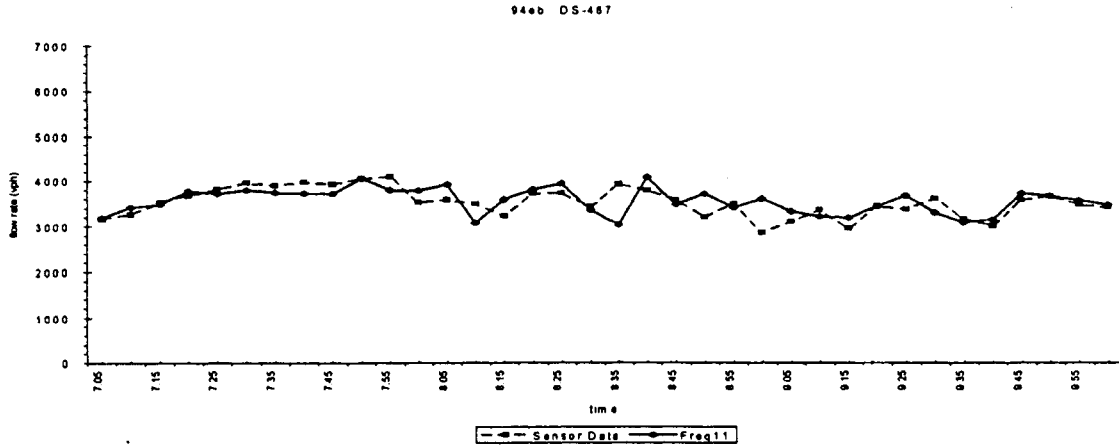


Figure 4.16 : Freq11 Results Compared to Collected Flow Data at DS 467

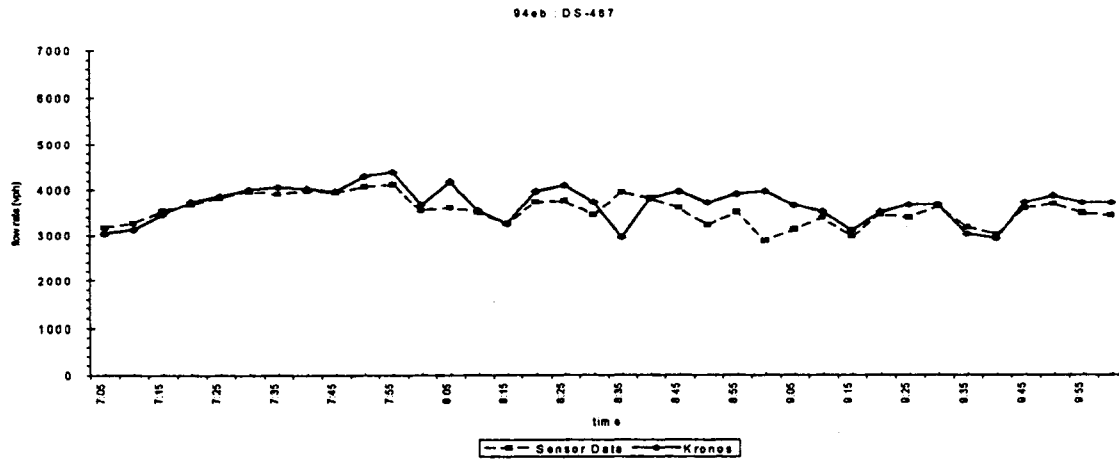


Figure 4.17 : Kronos Results Compared to Collected Flow Data at DS 467

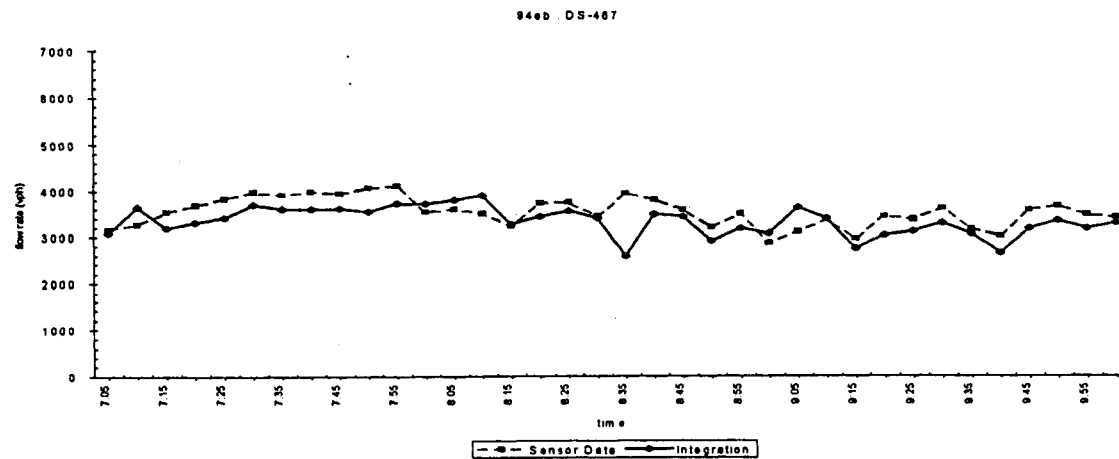


Figure 4.18 : Integration Results Compared to Collected Flow Data at DS 467

#### **4.1.4 Analysis of Quantitative Testing Results**

The average percentage differences for all the models tested were below 10 percent. It was noticed that all the models performed better with shorter freeway sections than with long sections. It can be seen that the differences were accumulated over distance and time with all the models. The FRESIM and FREFLO models required two separate files in order to simulate the three hour time period. The FREQ11 model treated each five minute time slice as a separate entity. Thus these models were able to reset any accumulated differences. It should be noted that the data used in this testing consist of only volume and occupancy. The direct measurements of speed data were not available because of the detector configuration at the test sections. Further, the calibration of the parameters for each model was conducted by using the trial-and-difference method, which indicate the possibility of getting better results if an extensive effort was made to determine the best set of parameters for each model for the test cases. Due to these limitations, and the relatively small amount of the real data used in this testing, any conclusion regarding the accuracy of each model can not be drawn. The results of this testing have reaffirmed the needs for better quality data and an efficient procedure to calibrate simulation models. The values of the simulation model parameters used for each case are included in the Appendix A.

#### **4.2 Qualitative Test Cases**

Five qualitative test cases were designed to measure spillback lengths and congestion dissipation times. Spillback lengths and dissipation times were measured using the densities operating within the congestion side of the flow-density curve. Spillback lengths were determined to be the maximum length of the congested section, i.e., the section with congested densities during the simulation. Dissipation times were measured from the time slice of maximum spillback to the time slice in which all densities are operating at uncongested levels. For the microscopic models, including Integration, that use stochastic distribution with random generation of vehicles entering the test freeway section, the average results of six runs were used for each test case.

#### 4.2.1 Case 1: I-494 Westbound at TH-100 Interchange

The table below shows the spillback lengths, release times, and where the spillback began using the geometrics of I-494 westbound at the highway 100 cloverleaf.

	FRESIM	AIMSUN2	FREFLO	FREQ11	KRONOS	INTEGRATION
spillback length (ft)	0	2985	4410	9900	2225	13920+
release time (min)	0	6	6	10	4	16+
spill at drop exit	no	yes	yes	no	yes	yes
spill at weave 1	no	yes	no	yes	no	yes
spill at weave 2	no	no	no	no	no	yes

Table 4.4 : Spillback Lengths and Release Times for I-494wb at the TH-100 Cloverleaf.

Table 4.5, table 4.6, and table 4.7 show the difference in the hourly flow rate between the beginning and ending points for the three subsections in the test section. With the macroscopic models adopting deterministic approach, it is expected that the difference between the flows entering and exiting each subsection is almost zero except the transition periods when the level of input flows changes. However, for those models adopting random vehicle generation approach with stochastic distribution, the flows entering and exiting each subsection can not be expected to be same. As indicated in the following tables, the microscopic models show significant variations for each time slice, while all the models show the expected pattern. The results from FREQ11 were not included in the table, since it simulates each time slice as a separate entity with the balanced input/output flows for each segment.

TIME	FRESIM	AIMSUN2	FREFLO	KRONOS	NTEGRATION
1	5	-30	30	12	-15
2	-85	-80	0	0	-40
3	-35	-5	30	0	5
4	-50	-105	150	102	95
5	-50	-120	-60	0	-75
6	-135	-305	0	0	85
7	885	785	1470	834	995
8	-75	-165	60	0	-20
9	-65	-10	0	0	-30
10	-25	-190	-180	-102	35
11	40	-225	30	0	-55
12	15	-335	0	0	755
13	-1325	-1030	-1890	-1152	-1520
14	-45	60	-30	0	-540
15	-20	-160	0	0	-5
16	-60	-30	150	102	90
17	25	-165	30	0	-205
18	-125	-85	0	0	30
TOTAL	-1125	-2195	-210	-204	-415

Table 4.5 : Difference Between Upstream Flow Rate and Downstream Flow Rate (vph) at Beginning of the I-494wb Test Section (600 ft to 5180 ft).

TIME	FRESIM	AIMSUN2	FREFLO	KRONOS	INTEGRATION
1	25	-75	-30	5	0
2	35	-25	0	-1	5
3	65	-70	0	-1	-70
4	5	45	-120	1	-60
5	130	135	150	-1	90
6	105	270	0	-1	-80
7	785	660	150	1180	585
8	475	1140	1260	233	1350
9	305	790	60	201	1330
10	45	1385	210	303	1295
11	145	1260	30	201	1035
12	310	1090	30	201	-585
13	-1030	-400	-90	-1311	355
14	-50	-1815	-1440	-1167	-335
15	85	-1905	-30	-1	-1310
16	70	-1655	-150	-1	-1430
17	40	-1045	120	-1	-920
18	165	-85	0	-1	-980
TOTAL	2660	-300	150	-161	275

Table 4.6 : Difference Between Upstream Flow Rate and Downstream Flow Rate (vph) at the Midsection of the I-494wb Test Section (5980 ft to 11090 ft).

TIME	FRESIM	AIMSUN2	FREFLO	KRONOS	INTEGRATION
1	20	105	-180	8	45
2	55	-45	-210	-10	-10
3	-125	35	-210	0	25
4	-85	-60	-210	6	-5
5	-185	-55	-330	-6	-105
6	30	165	-90	0	-20
7	200	50	-270	0	485
8	1195	355	360	-52	395
9	30	35	1020	0	-250
10	150	-190	120	0	10
11	40	-55	-30	0	-355
12	-170	55	90	0	115
13	-90	115	120	0	-700
14	-1405	670	120	74	45
15	-35	635	-720	0	-650
16	50	525	-60	6	-630
17	95	60	-150	-6	490
18	100	-60	30	0	445
TOTAL	-130	2340	-600	20	-670

Table 4.7 : Difference Between Upstream Flow Rate and Downstream Flow Rate (vph) at the End of the I-494wb Test Section (11090 ft to 16115 ft).

#### 4.2.2 Case 2: I-35W Southbound Through Burnsville

The table below shows the spillback lengths and the release times with the test case which has the same geometrics as the I-35w southbound section through Burnsville.

	FRESIM	AIMSUN2	FREFLO	FREQ11	KRONOS	INTEGRATION
upstream lane drop spillback (ft)	1750	3770	2755	n/a	3220	2475
upstream lane drop release time (min)	6	12	8	n/a	6	4
downstream lane drop spillback (ft)	3750	3300	4320	9920	5100	4100
downstream lane drop release time(min)	8	16+	12	14	6	12

Table 4.8 : Spillback Lengths and Release Times for I-35w Southbound Through Burnsville at Lane Drop Sections.

The following table shows the spillback lengths and the release times for the test case with an incident located at 500 feet downstream of the second lane drop for the same geometry.

ITEM	FRESIM	AIMSUN2	FREFLO	FREQ11	KRONOS	INTEGRATION
spillback length (ft)	8235	8185	12290	18515	11270	9185
release time (min)	10	20+	18	4	14	10
spillback occur incident	no	yes	no	no	no	no

Table 4.9 : Spillback Lengths and Release Times for I-35w Southbound Through Burnsville at Incident Section.

#### 4.2.3 Case 3: Hypothetical Geometry with Multiple Lane Drops

The table below shows the maximum queue lengths and the queue release times for the test case with multiple lane drop sections.

	FRESIM	AIMSUN2	FREFLO	FREQ11	KRONOS	INTEGRATION
spillback length (ft)	9500+	7400	9500+	8800	7900	9500+
release time (min)	14	16	20	8	12	16

Table 4.10 : Spillback Lengths and Release Times for the Multiple Lane Drop Freeway Lane Drop Sections.

The following table shows the maximum queue lengths and the queue release times for the same test section with an incident located at the halfway point of the two lane pipeline section.

	FRESIM	AIMSUN2	FREFLO	FREQ11	KRONOS	INTEGRATION
spillback length (ft)	6600	5200	8200	8800	7300	6600
release time (min)	10	12	16	8	12	10

Table 4.11 : Spillback Lengths and Release Times for the Multiple Lane Drop Freeway Incident Section.



## **4.2.4 Qualitative Performance Analysis**

### **4.2.4.1 Spillback Due to Congestion**

The first test to evaluate the capability of each model in treating the propagation of congestion used the geometrics of I-494 westbound at the TH-100 cloverleaf. For this test case, congestion was expected to start at the lane drop exit ramp. Little, if any, congestion was expected at the first weaving section and no congestion was expected at the second weaving section. Expected and estimated spillback lengths and dissipation times for the models can be seen in table 4.4. Three models did show densities large enough to suggest that congestion spillback was occurring at the first weave section. These were AIMSUN2, FREQ11, and INTEGRATION. Two models, FRESIM and FREQ11, did not estimate densities large enough at the lane drop exit to suggest that spillback conditions were present. INTEGRATION was the only model to experience congestion at the second weaving section. FRESIM model did not show any sign of congestion for this test. The estimated spillback lengths for AIMSUN2 and KRONOS were both less than the spillback lengths expected. In fact, both were at least 1900 feet shorter than the expected length. The FREFLO estimated spillback length was very close to the expected result. FREQ11 and INTEGRATION had very large spillback estimates. Dissipation times were close to the expected value for three models; AIMSUN2, FREFLO, and FREQ11. But, based on their spillback lengths, the AIMSUN2 dissipation time appeared to be too long whereas the FREQ11 dissipation time appeared to be too short. The KRONOS and INTEGRATION models had the dissipation times which were acceptable for their spillback lengths.

The second congestion test used the geometrics of I-35W southbound in Burnsville, Minnesota. Spillbacks were expected at both lane drop sections and the flow rates were designed to keep the two queues from overlapping. Values for the release times and spillback lengths can be seen in table 4.8. Aside from the FREQ11 model, these expectations held measured. At the first lane drop section, all of the simulation models showed congestion densities. All of the models, except for FREQ11, had spillback lengths close to the expected length of 2620 feet. Dissipation times were near the estimated value

of 4 minutes for all models except AIMSUN2 and FREFLO. At the second lane drop section, all models, with the exception of FREQ11, were within an acceptable range of the estimated spillback length of 3930 feet. FREQ11 had a queue length that did exceed the 7200 foot distance between the two lane drop sections. The dissipation times for most models were near the expected value of 10 minutes.

The last congestion test used a straight pipeline section consisting of a number of lane drops and the results can be found in table 4.10. Spillback was expected at the second and third lane drop sections and the queues from two locations were expected to overlap. This was the case for all models tested. Spillback levels were expected to be over 9500 feet long and require 14 minutes to clear. Three models did this; FRESIM, FREFLO, and INTEGRATION. FREQ11 was close to this expected spillback length at 8800 feet. AIMSUN2 and KRONOS were well below the expected spillback length. Dissipation times were within an acceptable region of the estimated value of 14 minutes. FREQ11 was the only exception, needing only eight minutes to release the spillback..

#### **4.2.4.2 Spillback Due to a Lane Blockage**

The first lane blockage test used the geometrics of I-35W southbound that passes through Burnsville, Minnesota. All six models recognized that an incident had occurred which blocked one lane. Results of the test can be seen in table 4.9. The two microscopic models had spillback lengths slightly larger than 8000 feet. The mesoscopic INTEGRATION was close to 9000 feet. The macroscopic FREFLO and KRONOS models had spillback lengths around 12000 feet. Finally, the FREQ11 model estimated a spillback length over 18000 feet long. The expected size of the spillbacks was 7900 feet. Dissipation times were close to expected value of 10 minutes for all but three models. AIMSUN2 exceeded the 20 minutes given for spillback clearance. FREFLO needed 18 minutes to release the spillback, but this value seemed appropriate when compared to the estimated spillback length. FREQ11 required four minutes to clear an 18000 foot queue.

The last lane blockage test was conducted using a straight pipeline which included a number of lane drop sections. Results of the test can be seen in table 4.11. All six simulation models captured the incident. As with the I-35W test case, the microscopic models predicted the shorter queue lengths. AIMSUN2 predicted a queue length of 5200 feet whereas the FRESIM and INTEGRATION models both predicted queue lengths of 6600 feet. Of the macroscopic models, KRONOS predicted the shortest queue length at 7300 feet. FREFLO and FREQ11 were both around 8500 feet. The expected spillback length was 6200 feet. All of the models, except for FREFLO, had release times close to expected eight minutes. The exception was FREFLO which required 16 minutes to release the spillback.



## 5. DEVELOPMENT OF A FRAMEWORK FOR AUTOMATIC SIMULATION ENVIRONMENT

In this chapter, a basic structure for an *automated freeway simulation environment* is developed including a conceptual design for an integrated database-simulation system and its requirements. This chapter will begin by briefly defining what is meant by an automated freeway simulation environment and outlining the practical motivations for developing such an environment. The basic framework for an automated freeway simulation environment will then be given, including descriptions of sub-systems and design recommendations. Finally, a prototype system was developed which incorporates some features of this design, and this prototype will be described in detail.

### 5.1 Introduction

#### 5.1.1 Definition

An *automated freeway simulation environment*, as defined here, is an integrated information system which simplifies the process of freeway simulation. This environment should help make freeway simulation a practical analysis tool for traffic/roadway-design engineers and researchers by:

- Automating intensive and/or repetitive processes.
- Providing intuitive and user-friendly software and/or hardware interfaces.
- Allowing the sharing of information between users and/or simulation applications.
- Integrating simulation with other useful information analysis tools.

#### 5.1.2 Motivation

Although freeway simulation has been in existence as a design/analysis tool since the late 1960s, it has not been greatly utilized by practitioners. One reason for this may have been a lack of credible software alternatives. Today, however, there are a number of viable freeway simulators available, many of which were reviewed and evaluated as a part

of this research. This evaluation process has helped to illuminate how these modern freeway simulators remain mostly impractical. The fundamental problems have to do with the access and use of information. Specifically,

- Collecting, storing, and correcting the various types of traffic data used in simulation can be a long and arduous task.
- Entering that information into a specific simulation tool can require many intense hours of effort.
- Once the data has been collected and entered correctly, there is often no easy way to share that information with other users and/or traffic applications (like another simulator, an incident detection system, a route-choice model, etc.).

This research, therefore, has attempted to design an environment which alleviates these problems.

## **5.2 Design Considerations**

### **5.2.1 Basic Structure**

The basic structure of an automated freeway simulation environment can be seen in figure 5.1. There are a number of sub-systems and concepts that contribute to this overall design, including:

- Data creation and collection
- Data storage and management
- Data distribution and access
- Simulation
- Data analysis

Each of these have unique design requirements for operation, and will be described in detail.

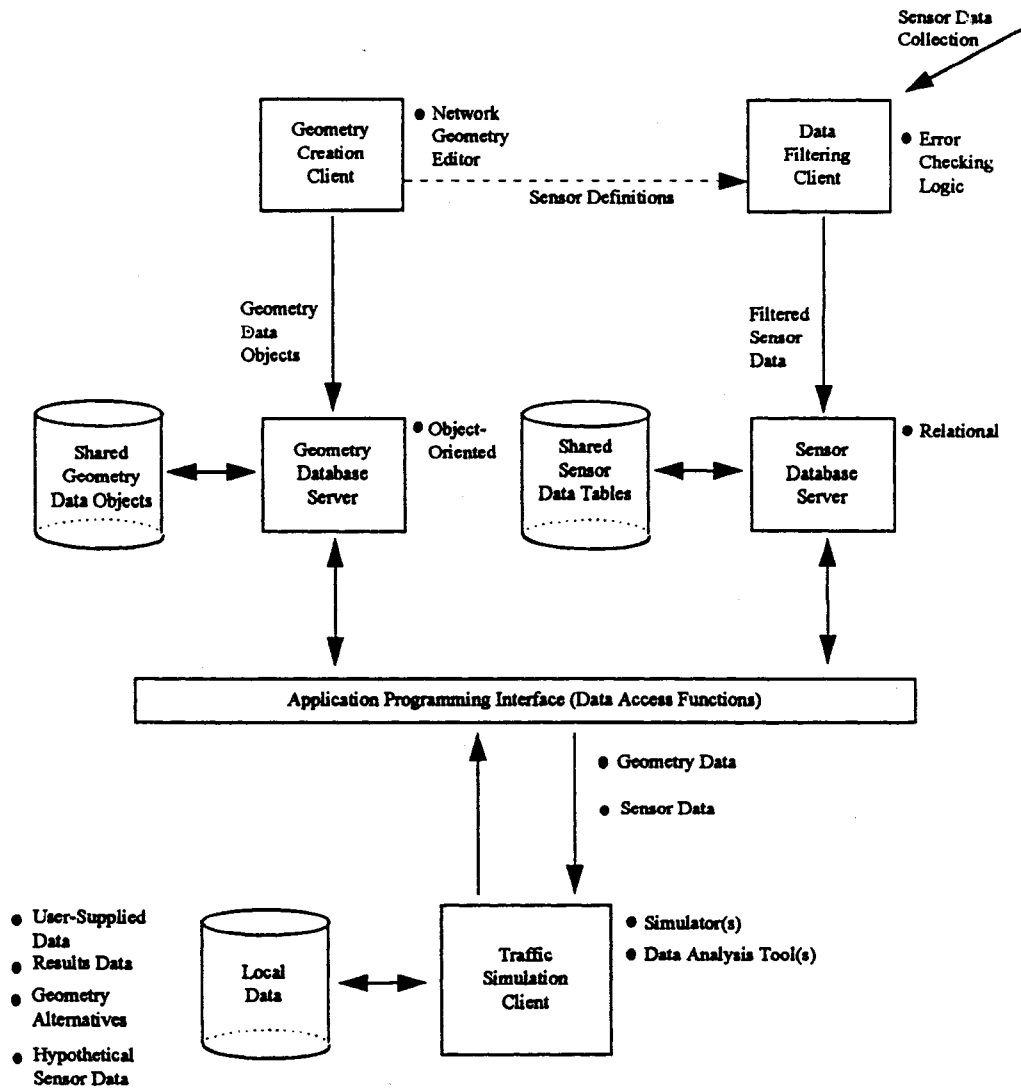


Figure 5.1 : Basic Structure of an Automated Freeway Simulation Environment

### 5.2.2 Data Categories

Before discussing the different components of an automated freeway simulation environment, it would be useful to enumerate the different types of traffic data that may be necessary for freeway simulation. Traffic simulation data can be partitioned into four major categories:

- Geometry data.
- Sensor data.
- User-supplied data.
- Results data.

Each of these data categories possess unique characteristics that affect how the data is collected, stored, managed, and accessed. A short description of each category is provided here. It is important to note that this research concentrated on the first two categories, geometry and sensor data, because they seemed the most natural candidates for automation.

#### **5.2.2.1 Geometry Data**

Traffic geometry data is any information pertaining to the actual physical layout of the traffic system of freeways and adjacent arterial streets that is useful to freeway simulation. For example, road sections, intersections, ramps, ramp meters, traffic signals, sensors, etc., are all physical “objects“ which have attributes and relationships with other similar and dissimilar objects. The collection of these objects, along with their attributes and relationships constitute the traffic geometry data. All freeway simulators require this information in one form or another, and in differing levels of detail.

#### **5.2.2.2 Sensor Data**

There are many different types of sensors that “detect” vehicles on freeways and make measurements pertaining to traffic flow characteristics. One common type of sensor is a *magnetic loop detector*, which is physically installed in the pavement of a road. A loop detector uses a magnetic field, or “bubble” protruding above the road’s surface to detect vehicles entering or leaving the field. The Minnesota Department of Transportation (Mn/DOT) currently uses “loops” in the Minneapolis/St. Paul freeway system as its primary method of gathering data (volume, occupancy, and speed) for traffic analysis and management.



Another type of sensor which has gained increasing interest in the ITS community is the *machine-vision detector*. Machine-vision detectors use video cameras and signal processing technology to “see” the road and detect vehicles. This sensing technique is typically more flexible and accurate, and can provide many more types of traffic measurements, such as information by lane and measures of effectiveness (MOEs). However, the technology is new and not yet widespread. Mn/DOT and the University of Minnesota (U of Mn) have collaboratively built a machine-vision test-bed along the newly constructed Interstate-394 in Minneapolis. This “I-394 Lab,” with operations located at the Center for Transportation Studies (CTS) ITS Laboratory utilizes 36 cameras placed strategically along a seven mile (3.5 miles X 2 directions) stretch of I-394 (see figure 5.2). The I-394 Lab will help provide new insights into the usefulness of machine-vision technology for data collection and traffic analysis.

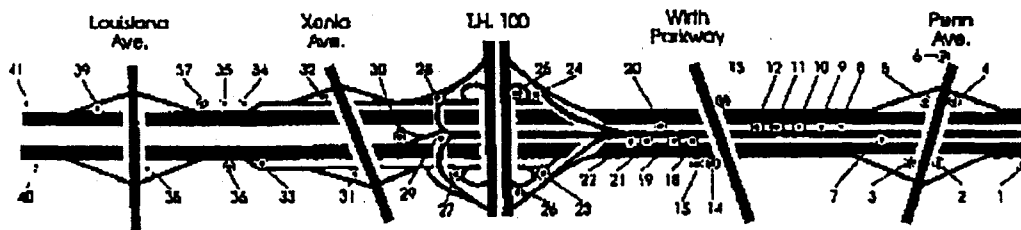


Figure 5.2 : Interstate-394 Laboratory cameras

Regardless of type, modern traffic management centers (TMCs) typically collect traffic data from sensors at periodic intervals (the TMC in Minneapolis receives sensor measurements every 30 seconds, and an additional aggregation every 5 minutes). This information is crucial for conducting traffic simulation based on real traffic conditions. Specifically, sensor data provides the flow information at the boundaries of the freeway network being considered. This information can also provide reality benchmarks for assessing the accuracy and usefulness of a given traffic simulator or model.

### 5.2.2.3 User-Supplied Data

There are a number of types of user-supplied data, and they differ considerably from simulator to simulator. Some examples include control plans, calibration parameters,

speed-flow-density relationships, probability distributions, time intervals and boundaries, and file/data management commands.

#### **5.2.2.4 Results Data**

The results of traffic simulation will again have a variety of types and formats, and are highly dependent on the specific simulator being used.

### **5.2.3 Data Creation and Collection**

#### **5.2.3.1 Geometry Data Creation**

In general, it is important to both the researcher and practitioner that the task of creating geometric information be as intuitive and easy as possible. Towards that end, a system utilized for geometry data entry should:

- Require minimal effort.
- Be map-based.
- Use a generic structure.
- Be object-oriented.
- Utilize graphical interfaces.

#### *Minimal effort*

Ideally, the process of geometry data entry should be fully automated, perhaps by using existing geographic/map-based data storage mediums. At the least, the user should never be required to recreate this information after its initial creation. In addition, the tool used to create this information in any medium should be user-friendly and intuitive, to minimize the time and effort required for this step.

### *Map-based*

The geometry creation tool should be map-based, providing a “real-world” look and feel to the user. In addition, it should have the ability to import other map images, like CAD files, to serve as a background guides or templates.

### *Generic structure*

The geometric design of the traffic network should be as “generic” as possible, to maximize the usefulness across multiple simulators and other analysis tools. A link-node structure is probably the most general, and intuitive.

### *Object-oriented*

The system should be “object-oriented” in its look, feel, and functionality. In other words, objects on the screen, such as sections, intersections, ramps, etc., should represent actual objects in reality. In addition, the user should be able to easily navigate among objects and multiple layers of complexity by using mouse clicks and simple keyboard commands. This intuitive representation of reality should ease the process of geometry data creation and manipulation. (Ideally, it would be written in an object-oriented language to aid future development.)

### *Graphical manipulation*

The geometry creation tool should facilitate graphical manipulation of the geometric characteristics, like drawing a new road, adding a lane, copying a sensor from one section to another, etc.

### *Recommendations*

A preliminary investigation was undertaken as part of this research to determine the feasibility of using a geographic information system (GIS) as the geometry creation tool and storage medium for freeway simulation. GIS software is designed to specifically handle (store, retrieve, manage, analyze) information referenced by geographic location,

which implies it may be useful as a simulator interface. Indeed, different GISs have been successfully implemented and applied to many other fields outside of transportation, and software vendors have been incorporating the capabilities to manage and analyze network and linearly referenced data common to transportation and traffic systems. An ideal situation would exist if existing GIS technologies could store and manage the geometry data required for traffic simulation.

Unfortunately, the determination of this research was that GIS technology is not yet suitable nor capable of providing the information needed for traffic simulation. The reasons for this are:

- GISs are based on geometric shapes, like lines, arcs, and polygons, and not on intuitive objects, which can have considerably more complex structures. For example, a road has a direction, which some GISs could model as a directed arc. However, the road also has width, like a rectangle. It might be possible to “force” this information into a GIS, but it would be too cumbersome and nonintuitive to be of much use.
- GISs are based on a relational database model, an excellent storage construct for many kinds of data, but a poor choice for geometric data. The relational structure requires a lot of lengthy searches that make GISs extremely slow and unsuitable for traffic simulation.
- GISs organize information into “layers.” For example, a map could be constructed which has a hydrology layer, a soil layer, and a pavement layer. (Some GISs handle the third-dimension more explicitly, but that is intended more for heights of buildings, structures, etc.) Using this scheme, the traffic network should all be a part of the same logical layer, especially since it is all connected. However, roads often travel over and under one another. There is no explicit way to indicate overlapping in a GIS because only one object per location is allowed per layer.

For these reasons, it was decided that GISs are not yet suitable for managing the geometric information used in traffic simulation. There is a movement in the industry to

develop standards for a GIS for transportation (GIS-T), which should help the situation. In addition, new GIS technology is being developed which is object-oriented. An object-oriented GIS should be much better suited to handling the complexity of information required by traffic simulators.

Until better alternatives emerge, it is recommended to use an existing freeway geometry creation tool that meets as many of the above requirements as possible. Since each freeway simulator has its own unique way of representing the geometric design of the traffic network, it would be impossible to find a general enough structure to handle all of this diversity. It is recommended, therefore, that the most general structure be chosen, and all tools that need geometric information (simulators or otherwise) would need to conform to that structure. This may mean modifying some software, but the advantages of having a unique shared representation should outweigh these setbacks.

#### **5.2.3.2 Sensor Data Collection**

The sensor data collected and used for simulation has three basic requirements:

- Detail.
- Accuracy.
- Flexibility.

#### *Detail*

At the basic level, freeway simulation requires the flow conditions at the boundaries and interchanges of the traffic network. To be done most correctly, however, a number of other measurements are needed. For example, vehicle classification information is needed by simulators that allow multiple vehicle types. Additionally, information collected by lane, speeds, lane-changing and weaving behavior, vehicle tracking for origin-destination mapping, queue lengths at ramps and surrounding arterials, congestion and incident information, and acceleration/deceleration rates could all be important data values for simulation. Equally as important, detailed data is needed for creating better models and benchmarking existing ones

### *Flexibility*

For much of the same reasons, flexibility is also an important requirement of sensor data used in simulation. More specifically, it would greatly aid freeway simulation in research and practice to be able to:

- Change detector locations and types.
- Add measurement specifications.
- Change collection time intervals.
- Use wide-area detection to determine initial traffic conditions.
- Densely pack sensors for fine detection of traffic flow dynamics.

### *Accuracy*

Simulation models, like most mathematical models, can be very sensitive to initial flow conditions. For this reason, it is very important for the sensor data to be accurate.

### *Recommendations*

The magnetic loop detectors used in the Twin Cities' freeway network are adequate detection devices for determining general traffic conditions. They are currently used to collect volume, occupancy, and speeds (the latter two are derived from the first) aggregated across lanes and two different time intervals (thirty seconds and five minutes). However, they do not meet the detail, flexibility, or accuracy requirements of traffic simulation modeling and practice.

Machine-vision sensors, on the other hand, are already capable of providing much of the desired information previously mentioned, and meet many of the detail, flexibility, and accuracy requirements of traffic simulation. For example, the AUTOSCOPE machine-vision detection device, initially developed at the U of Mn and deployed as part of the I-394 Lab, is capable of measuring vehicle presence and passage, vehicle speed and length classifications, vehicle flow rates, volumes, lane occupancies, and headways over time and level of service. In addition, newer versions of AUTOSCOPE provide MOEs, such as queue length. Appropriately spaced video cameras can also be used for wide-area detection of

initial traffic conditions for traffic simulation. Clearly, machine-vision needs to be seriously considered as an alternative to traditional sensing techniques, particularly as advancing ITS technologies continues to require more and more detailed, flexible, and accurate sensor data.

#### **5.2.4 Data Storage and Management**

Perhaps the most crucial element of an automated freeway simulation environment, or any automated information system, is the database management system (DBMS). In this context, the DBMS is responsible for managing and maintaining the data to be used in simulation other types of traffic analysis. The DBMS is the *hub* of information exchange between users/applications in a research or traffic management center. The design of the DBMS for automated simulation has the following features:

- Client-server architecture.
- Data filtering.
- Data separation.
- Data backup.
- Local data management.

##### **5.2.4.1 Client-Server Architecture**

Figure 5.3 shows a simplified view of a client-server network architecture.

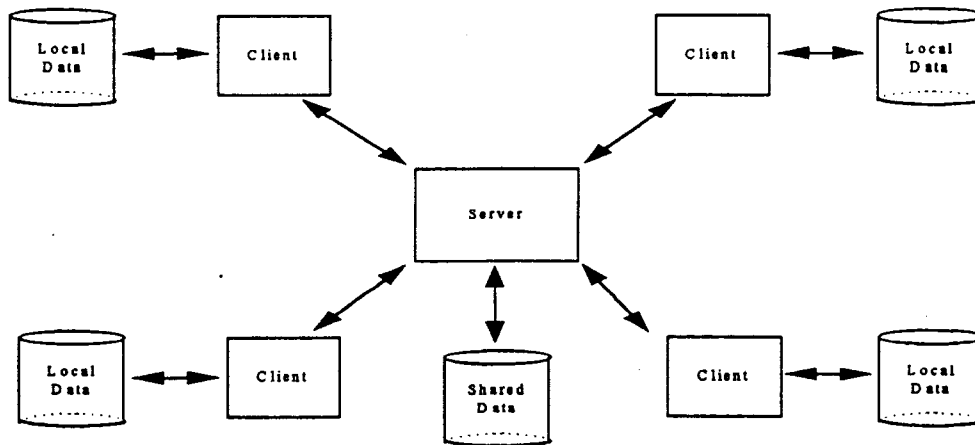


Figure 5.3 : Client-Server Architecture

It is assumed that any information system that would consider implementing pieces of an automated freeway simulation environment is already constructed in a client-server fashion. With this type of architecture, multiple machines, or “clients,” access another common machine, or “server,” to share information across a network. Servers are typically set-up to serve a particular purpose. For example, there could be a network server for managing network communications. A database server, on the other hand, is for storing and managing data. Clients make requests to the servers, and the servers “serve” the clients’ requests. There are many advantages to this configuration. To highlight a few:

- Individual client machines are released from the tasks that the server(s) take care of. This releases the local processor for a client’s individual needs.
- There is no need to replicate software and information on every client machine, because it is all located on the server.
- Multiple clients can share information, thereby integrating their work processes and increasing efficiency.
- Maintenance and information backup is localized to the server machine(s).



A database server has the following specific responsibilities:

- Holding the DBMS files and software.
- Reading from and writing to storage media (like a hard disk).
- Performing data set manipulations, like intersections, unions, exclusions, etc.
- Performing basic error/parity checks.
- Managing data access from multiple users, so that more than one client do not update the same set of data at the same time.

DBMSs are constructed to handle these responsibilities in efficient ways, which removes much of the work from individual clients.

#### **5.2.4.2 Data filtering**

A too often neglected aspect of a DBMS, particularly when data is collected from sensors or other mechanical devices, is the processing of checking for obvious errors in the data and correcting them where possible. The logic for this error checking can be quite complicated; nevertheless, it is an important aspect of a good information system. The sensor data is particularly prone to this - a broken or malfunctioning detector device can produce very anomalous simulation results. There are three options as to where this error checking should occur.

- Have the clients filter the data when they query it from the database.
- Have the server filter the data as part of its routine management tasks.
- Have a separate machine whose sole responsibility is to filter the data before it is sent to the database server.

The first option eliminates one of the advantages of having a centralized database server: removing management tasks from individual clients. The second is a much better option, because it reduces the number of filtering processes to one and relieves the burden on the client. However, if the load on the server is high (meaning that there are a lot of requests for data) the server could be spending too much time filtering and not enough time serving. For a non real-time system, as we are considering, the best option is the third. Here, a separate machine can filter the information before it reaches the database server,

freeing the server and the other clients from extra work. This is the option that was shown in Figure 1 at the beginning of this chapter.

#### 5.2.4.3 Data separation

The data used in freeway simulation can be separated and spread across multiple database servers (or disk/processor pairs in a parallel machine) depending on its characteristics. An individual server can then be customized to handle its particular type of data most efficiently. Data separation also reduces the user request load on a particular machine. A suggested data separation scheme is summarized by the tree in figure 5.4.

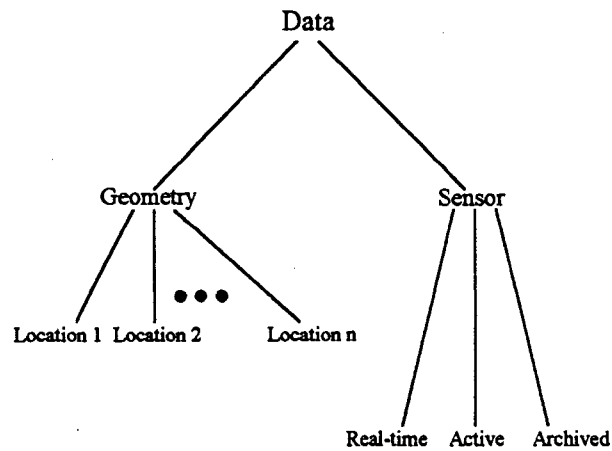


Figure 5.4 : Data Separation Tree

#### *Separation by Category*

A distinction has already been drawn between geometry and sensor data, and it is also recommended that these two data categories be partitioned to take advantage of their unique features. The geometry data is inherently object-oriented, with a complex structure and real-world objects corresponding to different data types, and an object-oriented database management system (ODBMS) should be used to manage this data. ODBMSs are particularly good at managing complex structures with many layers of

detail. ODBMSs are based on the tenants of object-oriented programming languages (like C++), and therefore provide excellent development environments as well as standard data management and access features. ODBMSs are relatively new technologies, which means that many were developed to be client-server based from the beginning. As with any new technology, some training and learning time is required to use ODBMSs, but the object approach is so intuitive and straightforward that any lost time will be quickly overtaken by increased productivity.

Another very important feature of this design is that an ODBMS is *extensible*. For example, assume that a freeway network has been completely built using the standard geometry creation tool used in the automated simulation environment. This network is saved in an ODBMS server as a set of objects. Then assume that a user is developing a new simulation application that requires the existence of freeway shoulders so that a more accurate representation of driver behavior around a stalled car can be implemented. If the geometry creation tool was not capable of specifying the existence of shoulders and the database was not extensible (like a file system, for example), then the new user could not use that geometry representation. An ODBMS, on the other hand, would allow that user to add a shoulder attribute to the standard freeway link object. Fortunately, any other existing software that does not need the shoulder attribute will still be capable of reading and writing to the ODBMS, without little or no software changes.

Sensor data is quite different from geometry data. Sensor data is temporal and highly dynamic, whereas geometry data is relatively static. The information from a sensor is reported in regular time intervals, like thirty seconds. For example, a loop detector station (ID #79) might report data that looks like the following:

ID #	Time	Volume	Occupancy	Speed
79	08:33:00			
79	08:33:30			
79	08:34:00			

Figure 5.5 : Example Loop Detector Station Report

It's not difficult to see that this data fits naturally into a tabular format. The DBMS which manages this type of data the best is a relational database management system (RDBMS). RDBMSs have decades of knowledge, research, and experience behind them, and are still the data management standard in most industries. These systems are highly "optimized" to provide good performance and have become much easier to use in recent years. They still lack some of the intuitive feel of ODBMSs, although vendors are adding more and more "object-like" features.

One detail with respect to this design is that the geometry and sensor databases must be connected in some way. For example, the loop detector station data entries above have ID #79. The geometry object that corresponds to this station must therefore also have ID #79 as one of its attributes so that the appropriate relationship can be established at run-time.

Machine-vision detectors, while offering far more of the features required for a simulation environment, have the added requirement of managing all of those features. How to best manage this data is an as-of-yet unanswered question, which the I-394 will hopefully help to answer. Preliminary thinking suggests some type of hybrid between object-oriented and relational DBMSs. It's also possible to imagine adding data objects designed specifically to manage machine-vision data. For example, one might define an "experiment" object that describes a particular user's detection scheme.

### *Separation by time*

The usefulness of sensor data, measured in terms of how often users will request it, will likely drop as time passes. This assumption implies that it would be good to separate data by time as well. The three "time databases" could be:

- Real-time data.
- Active data.
- Archived data.

*Real-time data* access, management, and use is beyond the scope of this research, but it is clear that a separate system will likely need to be designed to handle such data.

Of the other two, *active data* refers to data that has been recently acquired and more likely to be queried by users. Whether one defines this as the previous hour, day, or weeks worth of data really depends on the capabilities of the system and the patterns of the most frequent users. A TMC will likely want a shorter active data set, while a research laboratory might want a longer one.

*Archived data* is sensor data stored in a historical database. The storage medium might be flat files, a tape drive, an optical disk, a writeable CD-ROM, or other such device. The suggested separation by time is summarized in Figure 5.6.

#### *Separation by location*

If necessary, a potentially useful way to partition geometry data is by geographic location. There are many possible approaches to take with this. The data could be distributed across servers according to cities, counties, origin-destination zones, freeways, or some other scheme. Which scheme is best is another unanswered question. This separation anticipates that a typical user will likely simulate freeways in a confined area rather than the entire network. Another client might desire to simulate or analyze another part of the network. If the geometry objects for these two areas are on different servers, the two clients is no need for them to “get in one another’s way.” This could speed processing and make the system more user-friendly.

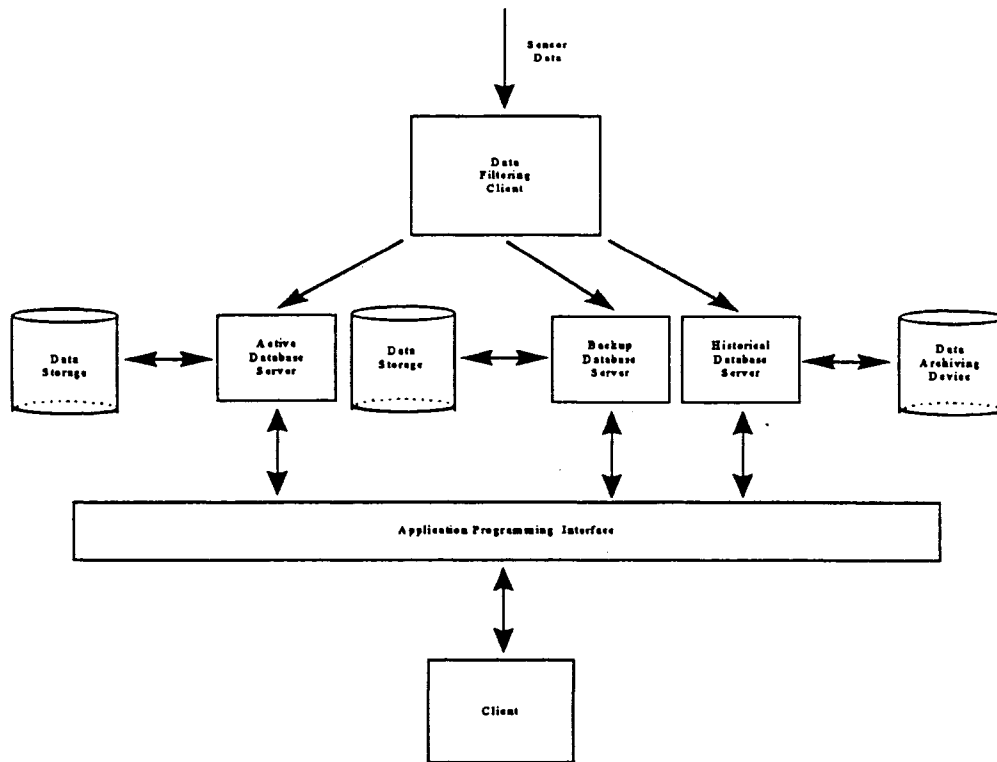


Figure 5.6 : Separating Sensor Data by Time

#### 5.2.4.4 Data Backup

There are two types of data backup plans that an information system designer should consider. The first is an *operational* backup plan, which is an alternative database server in case the primary server fails. This minimizes the chances of clients being unable to access data when they need it. The other type of backup plan is an *information* backup plan. It is recommended that a backup data storage device be used to periodically copy the data from the primary server. This storage device should be located off-site, so that a disaster or unrecoverable failure will not result in a complete loss of information. The information can be backed-up to the off-site location at non-peak hours, to minimize interference with users.

#### **5.2.4.5 Local Data Management**

There will be times when a user will want the option of storing and managing data locally on a client machine. For example, a roadway design engineer may be interested in simulating the effects of adding a new highway. Since this highway does not, and may never, exist, it should not be added to the centralized geometry database. Instead, the geometric information of the network surrounding the new highway location should be queried and stored locally. The user could add the new highway to that local geometry without effecting the central server. Similarly, user-supplied data, simulation results, and hypothetical sensor data may all need to be managed locally. Optimally, the data interfaces would be transparent to the user, meaning that they look the same regardless of whether they are using the central database or local files.

#### **5.2.5 Data Distribution and Access**

The following requirements are necessary to give users access to data stored in the database(s):

- Uniformity across clients.
- Transparency across servers.
- Graphics based query tool.
- Programming functionality.

##### **5.2.5.1 Uniformity Across Clients**

Every client in the client-server architecture should be able to access the database in exactly the same way. This ensures portability of applications across the network, and maximizes the potential of data sharing. In addition, users should not have to learn multiple access protocols to accomplish tasks. Object-oriented querying capabilities would help make this possible.

#### **5.2.5.2 Transparency Across Servers**

Assuming the data is partitioned across multiple servers according to the schemes described earlier, accessing data across these partitions should be as transparent as possible. For example, if there is a Minneapolis geometry database and a St. Paul geometry database, a user should be able to query the database for information about Interstate-94 from both databases without being aware of the separation.

#### **5.2.5.3 Graphical Query Tool**

Whenever possible, the querying tool used to make explicit data requests to the server(s) should be graphical, object-oriented, and map-based. For instance, a user should be able to point and click on a detector location and query data from a pop-up menu. This type of query tool greatly simplifies the Standard Query Language (SQL) commands that most query tools are built around, and thereby makes the database accessible to many more users.

#### **5.2.5.4 Programming Functionality**

It is particularly important that there exists a set of functions that a user can embed in a program, like a simulator, to query the database information. The reasons for this are obvious. Such a set of functions is known as an application programming interface (API).

#### **5.2.6 Simulation**

In the context of the aforementioned structure, the automated freeway simulator(s) should have the following attributes:

- Runs as a client process.
- Queries geometry and sensor data from central database servers.
- Conforms to generic geometric structure of geometry database.
- Intuitive map-based and object-oriented graphical user-interface.



- Sufficient local data storage and management power for handling user-supplied data, geometry alternatives, hypothetical sensor data, and simulation results.

Rather than expound on each of these points, the following example of a typical automated simulation session and accompanying pictures are provided.

0. Before beginning, it is assumed that someone has used the geometry creation tool to create the freeway network, and this geometrical structure has been saved as objects in the geometry database server. In addition, the sensor data has been collected and stored in the sensor database server as previously described.
1. The user opens the automated freeway simulation application (figure 5.7) as a client process in the client-server computer network. The user then requests to open a freeway network that is stored in the geometry database server. The application automatically queries the geometry database for a minimal set of information - to provide the user a “stick-figure” view of the freeway network.

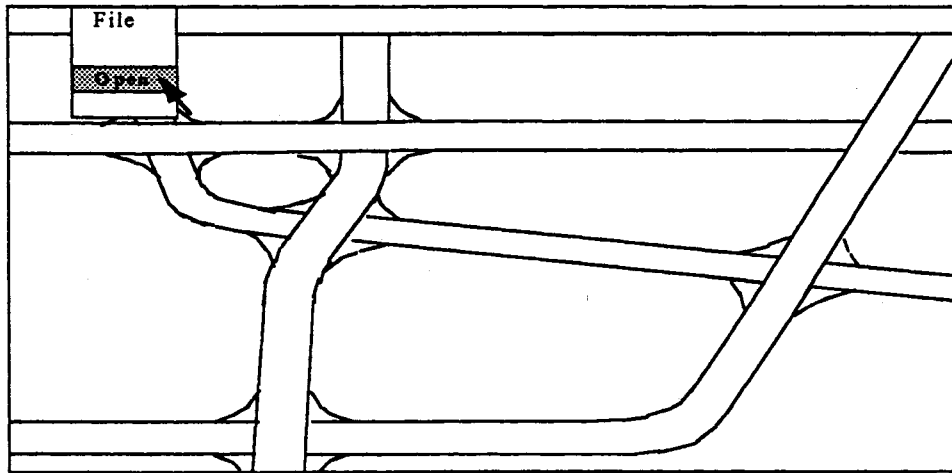


Figure 5.7 : Opening of Freeway Network File

2. The user will then choose a sub-network with which they want to work (figure 5.8).

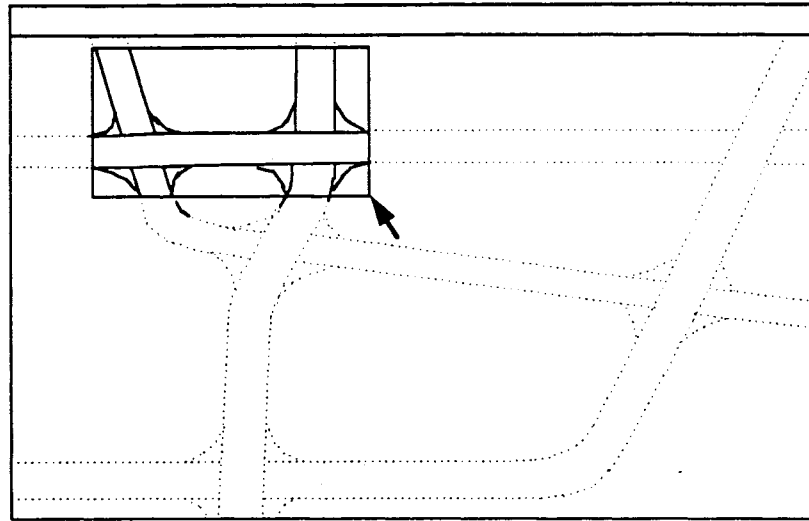


Figure 5.8 : Sub-Network Selection

3. Based on the specified boundaries, the simulator will automatically query the geometry database for the detailed information needed for simulation, and display that information graphically to the user (figure 5.9).

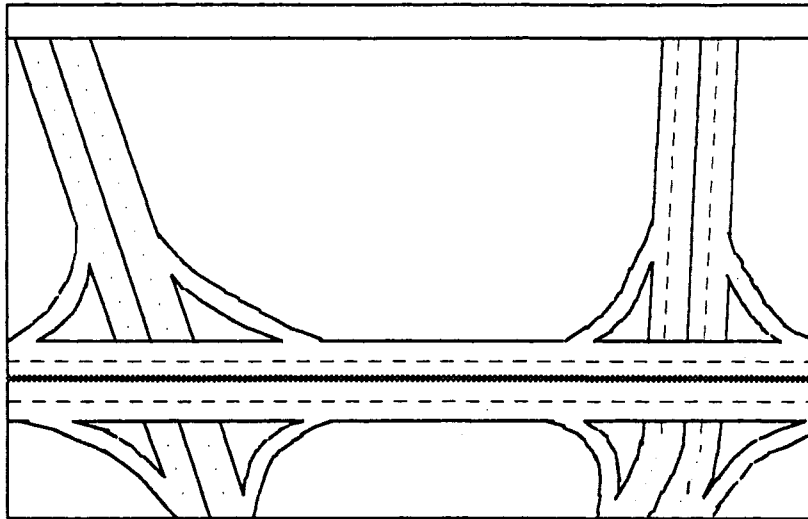


Figure 5.9 : Detailed Graphical Display

4. The user will then request to load the sensor data information to be used as initial flow conditions and/or arrival distribution parameters (figure 5.10).

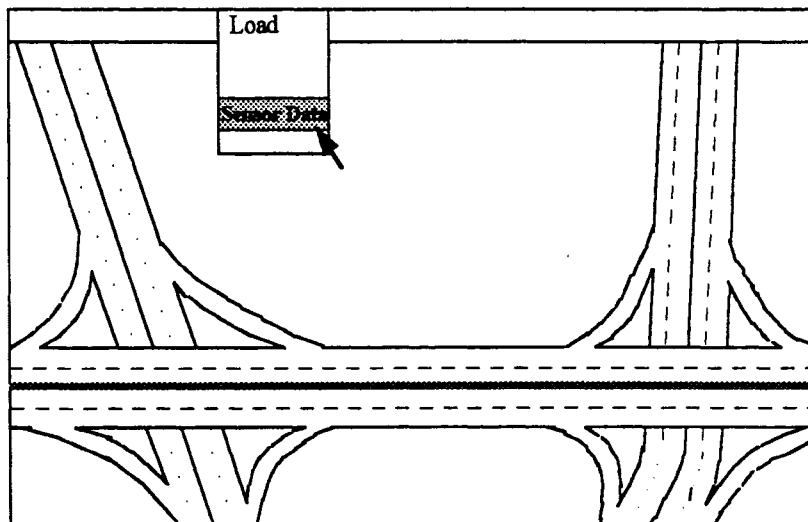


Figure 5.10 : Loading Sensor Data

5. A choice will then be made between using the central database server or sensor data stored locally (figure 5.11). (The local data could be hypothetical sensor information and would need to have the same basic structure as the data stored in the central server.) Assume the user chooses the central sensor database.

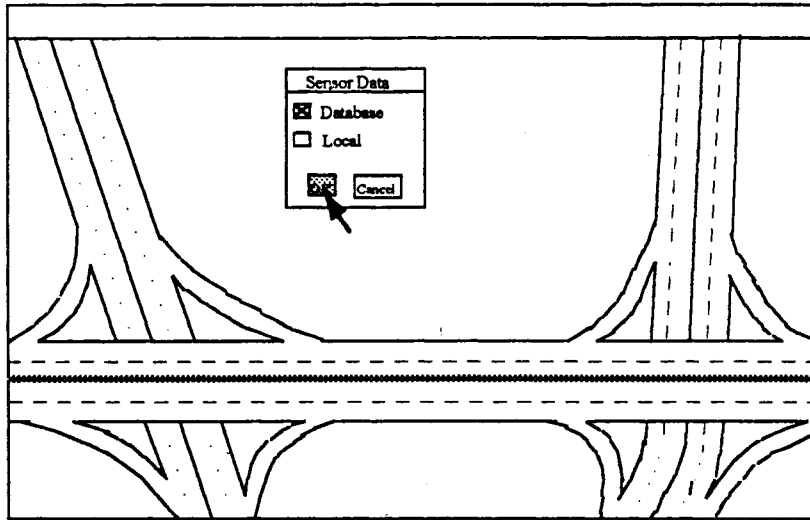


Figure 5.11 : Choosing Sensor Data Storage Unit

6. The user will then be asked to provide the sensor data query details, like the date, simulation start and end times, and the time-step intervals at which to query the data (figure 5.12).

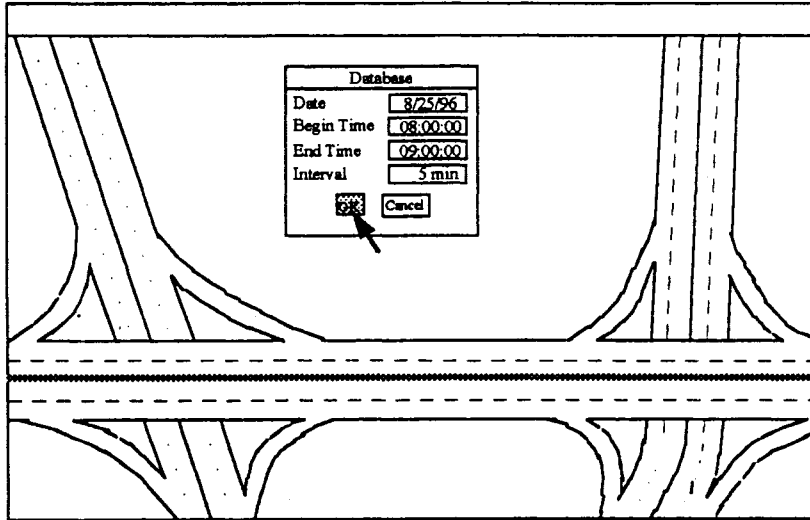


Figure 5.12 : Sensor Data Query

7. The simulation application will then use the geometrical information that it has already queried to specify the relevant sensor locations in the sub-network that has been specified (figure 5.13). This information will then be used to automatically query the sensor database for those particular sensors.

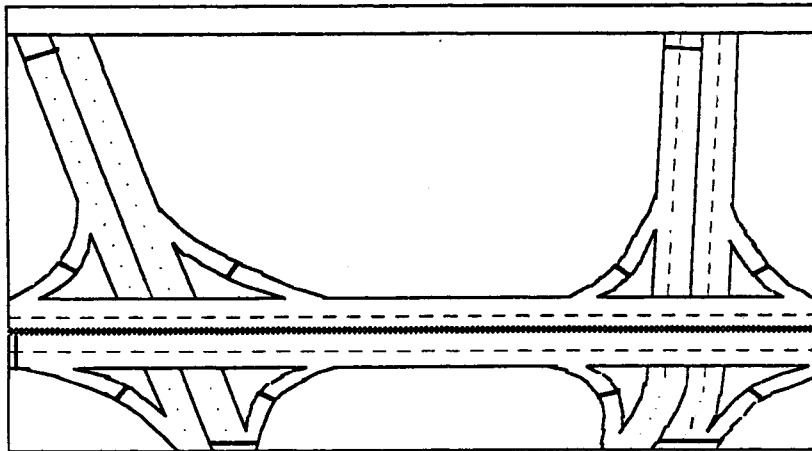


Figure 5.13 : Sensor Locations in Sub-Network

8. The simulation application can then be run (figure 5.14).

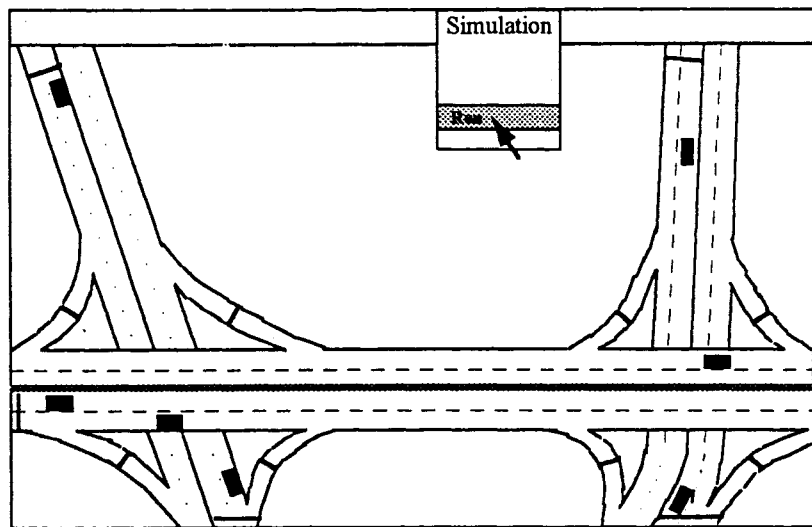


Figure 5.14 : Starting a Simulation Run

### 5.2.7 Data analysis

The final component of the automated freeway simulation environment is the data analysis functionality. Specifically, there should be the capability of analyzing the results of simulation runs. Two recommended tools for the automated environment are:

- A spreadsheet application which can easily import data and give users the capability of performing calculations, statistics, graphing, and other complex analytical techniques.
- A map-based tool which allows a user to graphically observe traffic conditions (densities, speeds, flows, etc.) change over time during the course of a simulation.

## **5.3 Prototype Development**

### **5.3.1 Motivation**

One of the freeway simulators evaluated as a part of this research, AIMSUN2, is already integrated into an environment that shares some of the attributes of the aforementioned design. This environment, GETRAM, was the basis for the development of a prototype automated freeway simulation environment in the CTS ITS Lab at the U of Mn. Specifically, an attempt was made to integrate the macroscopic freeway simulator developed at the U of Mn, KRONOS, into the GETRAM environment. The motivation for this development were:

- To practically demonstrate some of the concepts of the automated freeway simulation environment.
- To show the strength of a generic geometry by attempting to merge two very different simulators under a common geometric structure.
- To test the feasibility of using GETRAM as the foundation of an automated freeway simulation environment to be deployed in the CTS ITS Lab at the University of Minnesota.

### **5.3.2 Background**

#### **5.3.2.1 GETRAM**

GETRAM, or “Generic Environment for TRAffic Modeling,” was created at the Universitat Politecnica de Catalunya in Barcelona, Spain. It functions as an integrated environment by combining various models and tools for traffic analysis under a single platform. Its features include:

- A user-friendly traffic network editor with a close-to-reality appearance.
- Intuitive graphical objects with multiple levels of detail.
- The ability to load a CAD-type graphics file as a background template.
- The ability to choose sub-networks on which to run applications.
- A logical database design with a well-defined application programming interface (API).

- Multiple integrated traffic tools, including a microscopic simulator (AIMSUN2), an assignment model (EMME/2), and an adaptive control model (CARS).
- The ability for one tool to share data with another tool.
- Potential to extend the environment to include other tools.
- A results presentation user-interface.

GETRAM was written to run on an XWindows UNIX workstation, but can also be run on a PC under Windows/NT using the Exceed XWindows emulator.

GETRAM's functional design is focused around a common database for storing and managing traffic data, including network geometry information, traffic sensor data, and control plan information. Communication with this database is possible through an application programming interface (API). This API consists of a set of C functions that an application can use to add, change, or extract information from the database in a pre-defined manner. The API is an important feature of GETRAM because it specifies a "common language" through which all applications may communicate with the database, and therefore with one another. For instance, an assignment model can be used to specify a particular traffic state (entrance flows, turning percentages, etc.) which is stored in the database using the API. Then a simulator can use the same API to access this traffic state information and use it as input data for a simulation run. Having such a "common language" is of primary importance for creating an integrated environment.

GETRAM's functional design is represented in figure 5.15. Although GETRAM was written to be expandable to interface with a database server, it currently uses text files for data storage and management. The three data types, network geometry information, traffic sensor data, and control plan information, are stored in physically separate locations (specified as sub-directories in the UNIX file system), and have unique API functions for manipulating them. The developers of GETRAM refer to the network geometry portion of the data as the "GETRAM database," and the other two data types as "storage." Since the API accesses all three types of data, however, the set of their union more accurately represents the "GETRAM database."



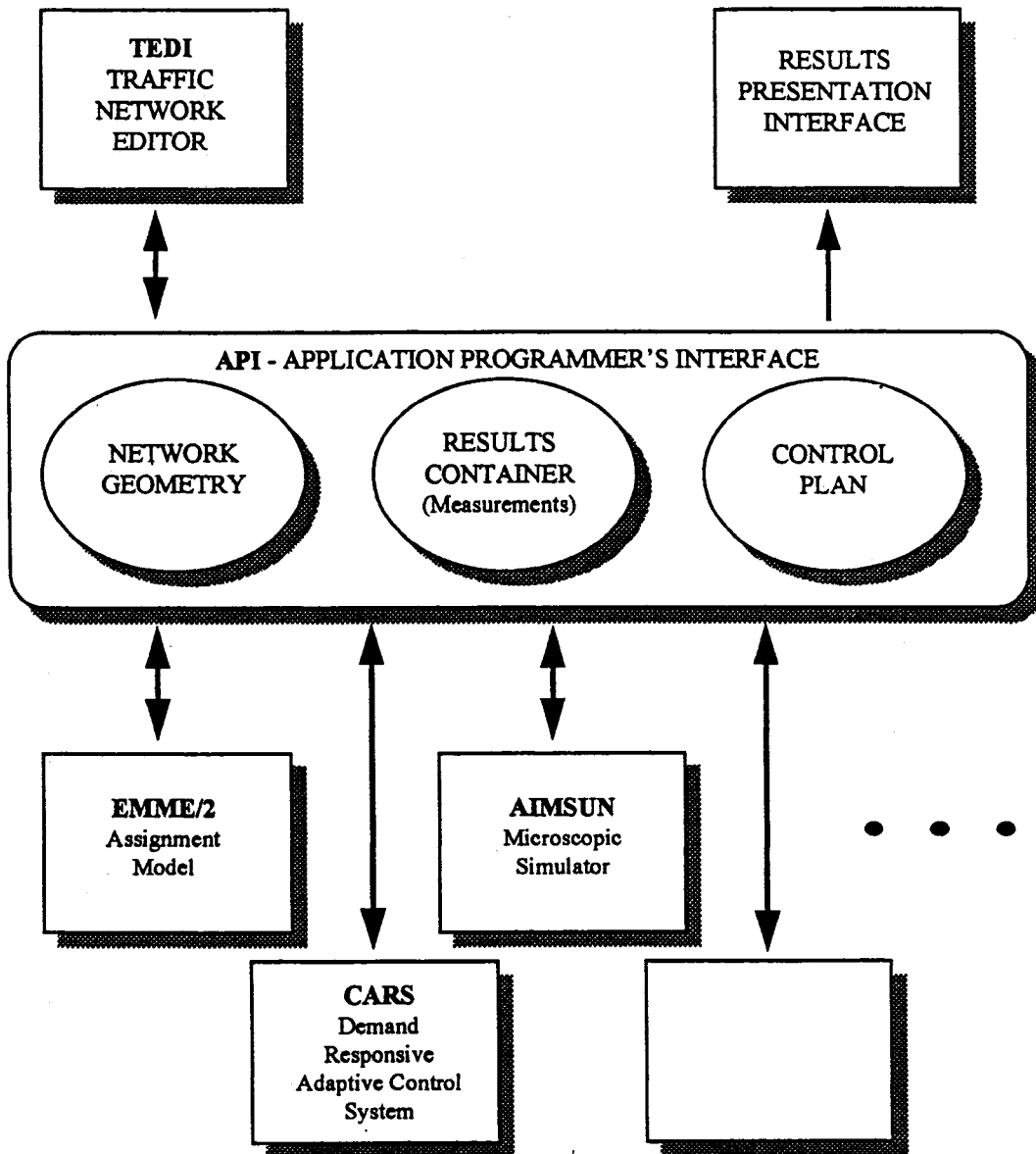


Figure 5.15 : GETRAM Functional Design

Of particular interest in this functional design is how the Advanced Interactive Microscopic Simulator for Urban and Non-urban Networks (AIMSUN2), a microscopic simulator also developed at the Universitat de Catalunya, is integrated into GETRAM. The process of simulation using AIMSUN2 involves drawing a traffic network, specifying traffic sensor information and control plans, and executing a simulation run. To accomplish this, the Traffic network EDItor (TEDI) is used for creating a traffic network.

The TEDI has a user-friendly graphical interface which utilizes intuitive objects like freeway links and nodes, and has a close-to-reality network appearance. Additionally, to assist the user in the detailed task of drawing a traffic network, a CAD-type (DXF) background image can be imported as a background template. Next, the TEDI is used to specify traffic measurement information and control plans, and all of the data is saved in the database. At this point, the AIMSUN2 application is opened, and the API (TDFunctions) is used to read the information from the database that it needs to execute a simulation run. (Note that the TEDI and AIMSUN are separate software applications, even though they have a very similar graphical appearance.) This information must then be “translated” into the format that AIMSUN2 uses to model and understand various kinds of traffic information. Once the translation is complete and some simulation specific information has been entered, AIMSUN2 can run a simulation. Upon completion, the results presentation interface can be used to view the results. Figure 5.16 illustrates the entire simulation process.

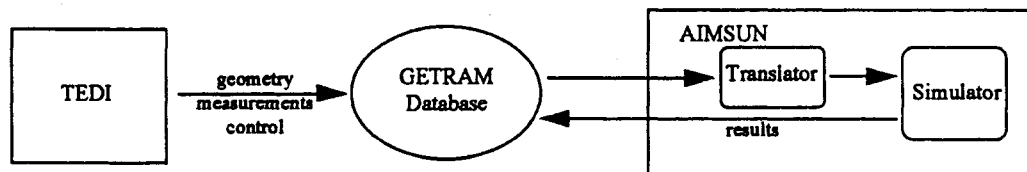


Figure 5.16 : AIMSUN2 Simulation Process

### 5.3.2.2 KRONOS

KRONOS is the freeway network simulator that was developed at the University of Minnesota. It uses a macroscopic, continuum flow model for simulating traffic flow. Functionally, KRONOS can be understood according to figure 5.17. In particular, there are three separate executable programs which are necessary to build freeway networks, run simulation, and view results: the Input Module, the Simulation Module, and the Output Module.

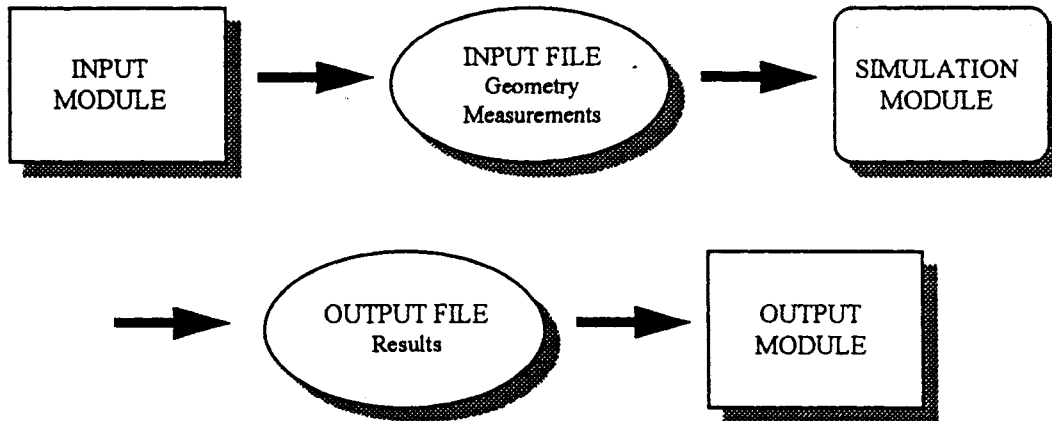


Figure 5.17 : KRONOS Simulation Process

The KRONOS input module is used to specify the traffic geometry and input sensor data and other simulator specific parameters. (KRONOS uses a geometric construct called a “segment” which is quite different from the link-node structure of GETRAM.) This information is then saved in a KRONOS input file. Once the input file has been created, then another application called the simulation module is run which performs the simulation run. Once completed, the output module can be used to view the results.

### 5.3.3 Geometric Design

#### 5.3.3.1 GETRAM

##### *Structure*

GETRAM’s network geometry database design is generally specified in figure 5.18 as a set of graphical/conceptual objects and their relative hierarchies.

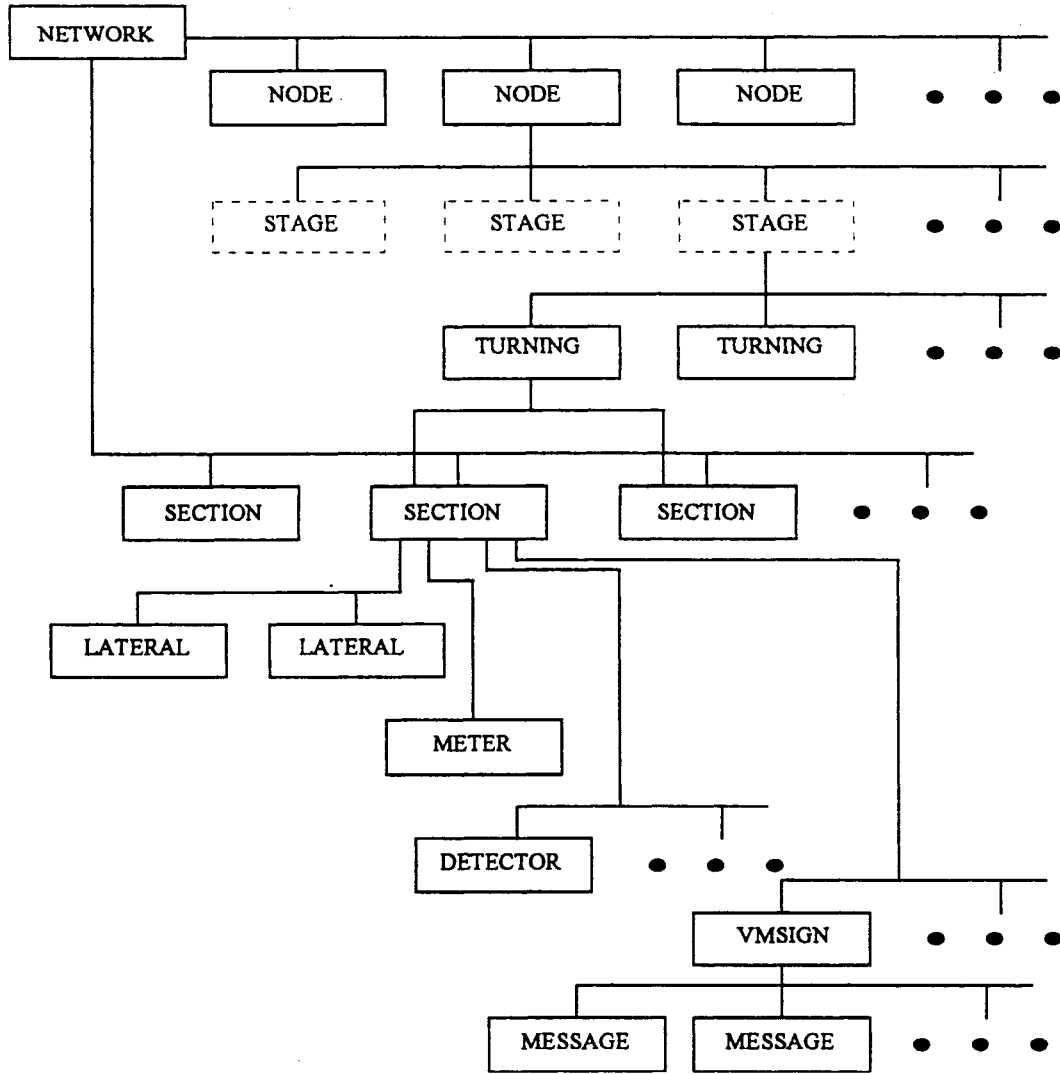


Figure 5.18 : GETRAM Geometry Structure

A GETRAM traffic network can most generally be understood as a set of various sized uni-directional roadway sections, connected in a particular way. As is clear in the diagram, a *Network* object consists of two primary object sub-types: *Sections* and *Nodes*. The design of these primary data objects are described in detail.

### Sections

*Section* is one of the two fundamental geometrical objects in GETRAM. Most generally, a *Section* can be understood as a straight piece of uni-directional roadway (freeway or arterial). Among other things, a *Section* is defined by its location, length, width, the number of lanes that run along its entire length, and the number and location of lateral lanes (lane add/drops). For example, a generic *Section* in GETRAM may appear as:

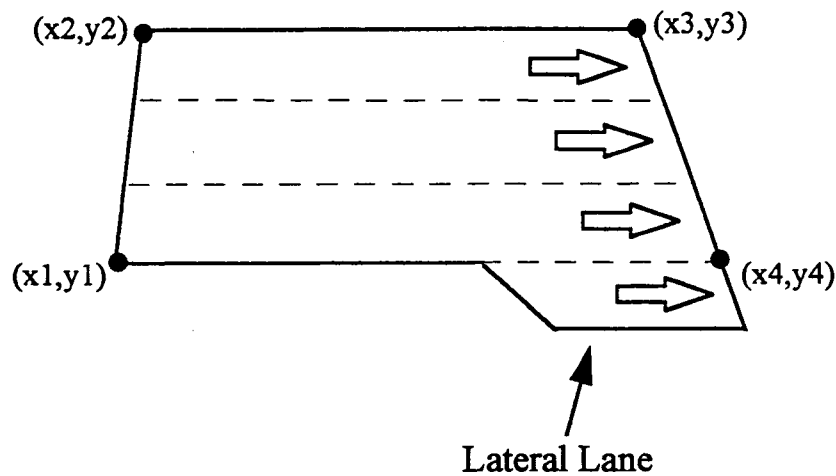


Figure 5.19 : GETRAM Section

Notice that this *Section* is specified by four two-dimensional coordinates. These coordinates specify not only the *Section's* location in coordinate space, but also its length, width, and direction. Not including its *Lateral*, the above *Section* has three “main” lanes. The detailed design description of a GETRAM *Section* and its attributes (table 5.1) is as follows:

Attribute name	Data type
Section_Id	Integer
Section_Name	String
Level	Integer
Number_of_Lanes	Integer
Stair_Exit	Boolean
Section_Type	Integer
Road_Type	Integer
Max_Speed	Real
Capacity	Real
Optional	String
x1	Real
y1	Real
x2	Real
y2	Real
x3	Real
y3	Real
x4	Real
y4	Real
Number_of_Laterals	Integer
Number_of_Detectors	Integer
Number_of_Meters	Integer
Number_of_VMSigns	Integer

Table 5.1 : GETRAM Design Description of a Section

Notice that each *Section* has a Level, which allows *Sections* to be stacked on top of one another in three-dimensional space. Note also, the “Stair\_Exit” Boolean value. This specifies whether the end of a *Section* terminates in a straight line, as in figure 5.19, or whether it is staggered like “stairs.” The stair design is sometimes found at arterial intersections.

*Laterals* in GETRAM are lane adds or lane drops. Besides *Laterals*, *Sections* can also have *Detectors*, *Meters*, and *VMSigns* (Variable Message Signs) associated with them. Each of these sub-objects has its own set of unique attributes.

### *Nodes*

The other fundamental geometrical object in GETRAM is the *Node*. A *Node* can be understood as the *location* where two or more *Sections* converge, and can be one of two types. The first is the spot where two or more freeway *Sections* are joined, such as the beginning of an on or off ramp. This is the simpler case, and is referred to in GETRAM as a *Juncture*. (*Junctures* can also be “fictitious” in the sense that they are often imposed by the modeller to create the illusion of curvature in the road. Since GETRAM does not support curved road *Sections*, two or more straight *Sections* can be linked to follow a curve’s general path.) An example *Juncture* can be seen in figure 5.20.

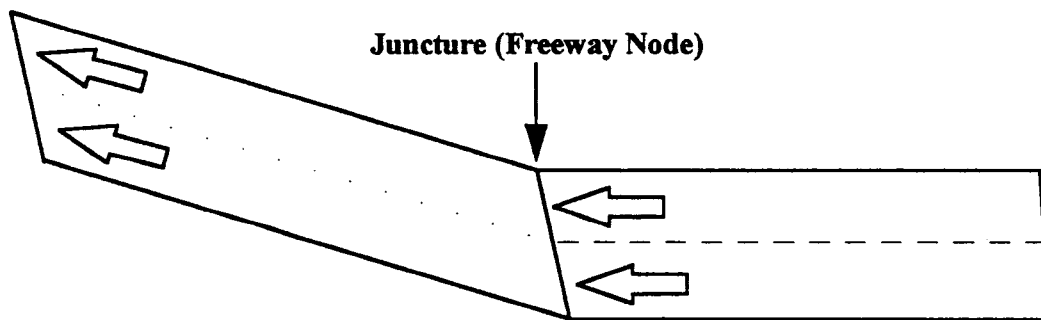


Figure 5.20 : GETRAM Juncture

The other type of *Node* is an arterial intersection, which GETRAM calls a *Junction*. (The use of *Juncture* and *Junction* is confusing and unfortunate.) A *Junction* can be considerably more complex than a *Juncture* because of the possible turnings and control specifications that can occur in an intersection. An example *Junction* can be seen in figure 5.21.

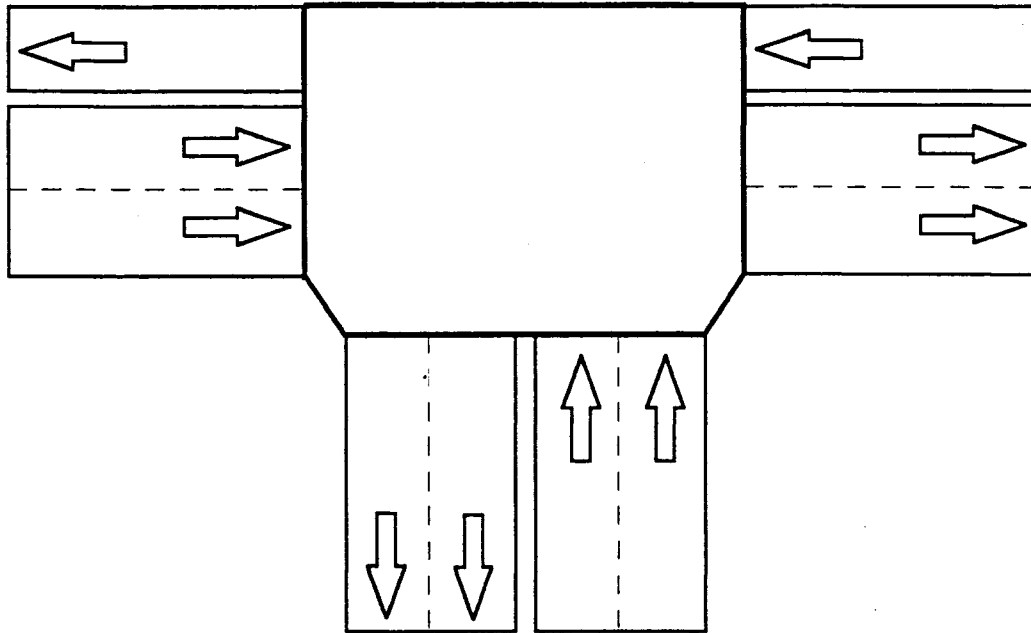


Figure 5.21 : GETRAM Junction

The detailed design description of a GETRAM *Node* and its attributes is as follows:

Attribute name	Data type
Node_Id	Integer
Node_Name	String
Node_Type	Integer
Number_of_Stages	Integer
Number_of_Turnings	Integer
Yellow_Box	Boolean
Zone_Id	Integer
Zone_Connector	Integer

Table 5.2 : GETRAM Design Description of a Node

*Stages* are the different “phases” that occur in controlling a Junction (intersection). For instance, there may be a “red” stage for a particular direction, then a “green” stage, etc..

*Turnings* are the objects that contain information of vehicles allowed movements in a



*Node*. In Junctions, a *Turning* can belong to both the *Node* and to one or more *Stages* of control, and is defined by the range of lanes from which a vehicle can move from one *Section* to another. A Junction does not have stage information, so *Turning* objects belong only to the *Node* in that case.

### 5.3.3.2 KRONOS

#### *Structure*

The structure of the KRONOS input data can be viewed in figure 5.22. A freeway network is composed of a set of one or more *Freeway* objects connected in a particular way. Each *Freeway* is further composed of a number of *Segments*. Depending on a particular *Segment's* type, it may also have other data objects associated with it, such as *On Ramp*, *Off Ramp*, *Weave*, or *Lane Add/Drop*. Finally, a *Segment* could have an *Incident* specified in it.

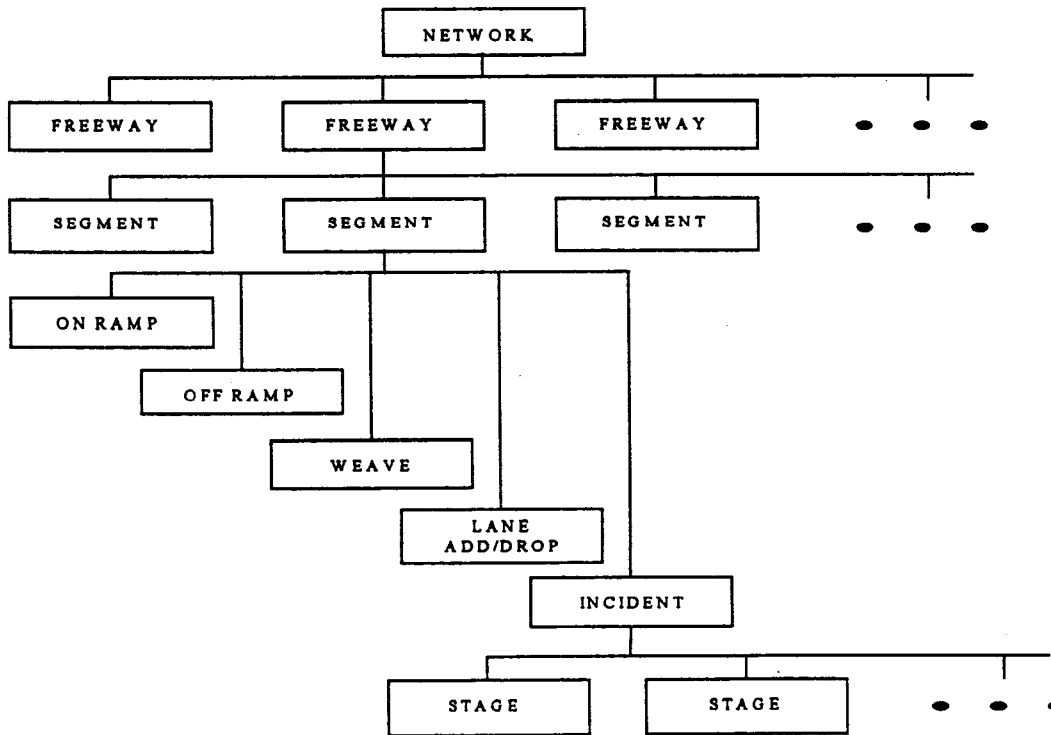


Figure 5.22 : KRONOS Geometry Structure

### *Segments*

Perhaps the most distinguishing feature of KRONOS is the way in which it builds freeway networks. In contrast to the “geometrical” approach used by GETRAM, KRONOS uses a “topological” approach when drawing a network. In other words, there are a certain set of “segment types” that are distinguished by features such as having an on/off-ramp or not. Two sections of roadway can vary greatly in length, the number of lanes, and speed limits and still qualify as the same segment type under the KRONOS scheme. Figure 5.23 offers a sampling of the 25 different segment types and their basic shapes. (Type 16, a collector-distributor road type, is not included because KRONOS does not currently work correctly for that type.) By stringing together many of these types, freeways are quite easy to build in KRONOS. They do not, however, have a realistic graphical representation.

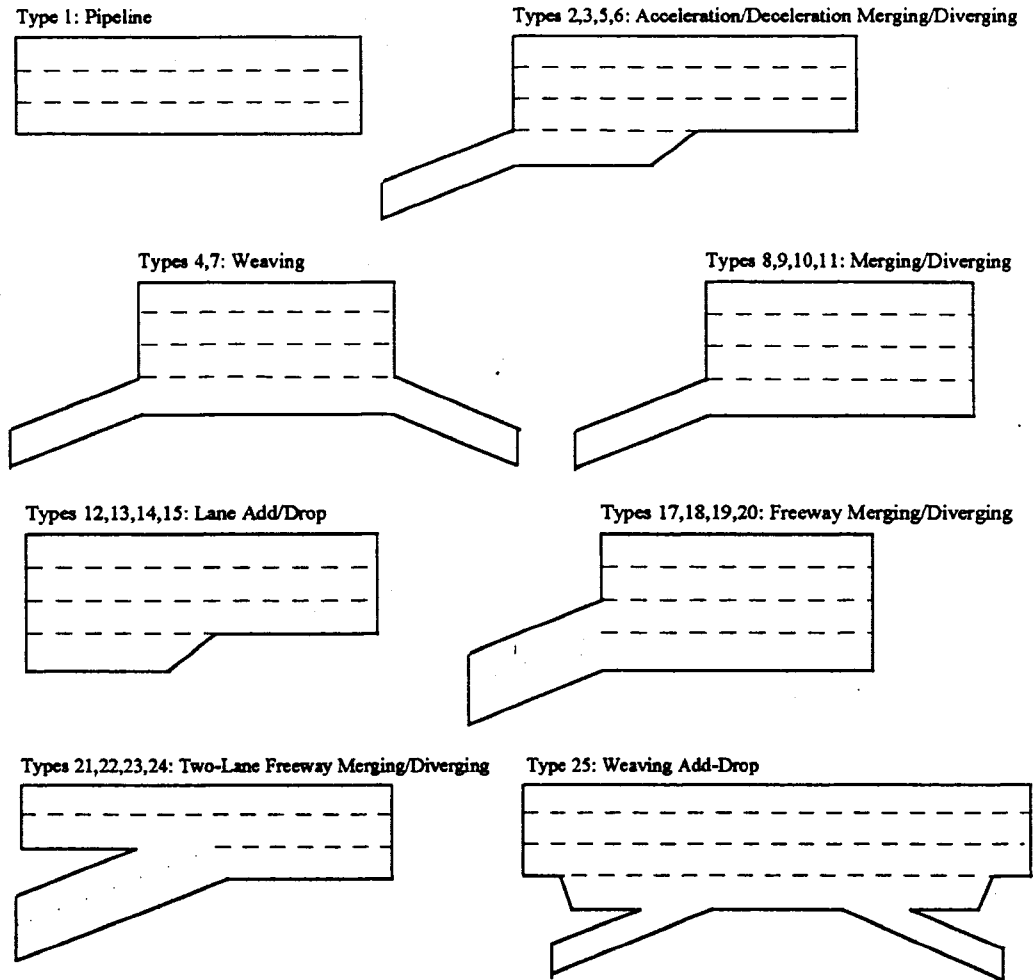


Figure 5.23 : KRONOS segment types

#### 5.3.4 Data Translator

Now that the basic design and data structures of both GETRAM and KRONOS have been presented, the prototype automated freeway simulation environment that was developed can be described. Since GETRAM was designed to be a “generic environment” for conducting various types of traffic analysis, a decision was made to explore the feasibility of integrating KRONOS into that environment. More specifically, an attempt was made to take a freeway system created using GETRAM and “translate” that information to create a KRONOS simulation input file. The successful implementation of

such a “translator” would support the previously outlined design in that a generic geometric structure would be used to inform multiple traffic analysis applications.

#### **5.3.4.1 Assumptions and Completion Tasks**

The primary challenge for the translator was to convert the link-node geometric structure of GETRAM into the segment types used by KRONOS. Considering the time and resource limitations, some simplifying assumptions were made:

- A single freeway would be used.
- All on and off ramps on that freeway would be straight segments.
- Some KRONOS segment types would not be covered (21-25,16).
- All network flow states would have equal time offsets.
- Only three GETRAM vehicle modalities would be used (Passenger, Light Commercial, and Heavy Commercial).
- All GETRAM sections would be built with capacities (not required for AIMSUN2).
- Any information that was not obtainable through GETRAM would have to be entered by the KRONOS user the after translation was complete.

In addition, it was assumed that a freeway had already been created in GETRAM (using TEDI) that adhered to the preceding assumptions, and was saved as a GETRAM network. Also, it was assumed that the network flow states that would normally serve as mean arrival rates in AIMSUN2 had been saved in GETRAM’s “Results Container” as part of the network as well. The translator’s tasks were then divided into the following sub-tasks:

1. Read the GETRAM network geometry information.
2. Determine the list of GETRAM segments that would be considered the mainline traffic segments for the freeway.
3. Create a list of KRONOS segment types by collecting various GETRAM segments together along the mainline.
4. Read the GETRAM flow information.

5. Convert the flow rates to KRONOS time step rates and combine them into the segment type data structures.
6. Write the new KRONOS segment types and attributes into the KRONOS input file format.

The writing of this translator was achieved using a Silicon Graphics, Inc. (SGI) X-Windows UNIX workstation with accompanying C compiler and the Motif graphics library. The translator takes advantage of GETRAM's Application Programming Interface (API), called "TDFunctions," to follow the preceding sub-tasks. In fact, the translator itself was written as an addendum to the TDFunctions, creating new TDFunctions specifically for KRONOS translation. One of the freeway sections created as a part of the testing of AIMSUN2, westbound Interstate-494 west of the St. Paul-Minneapolis airport, was chosen as a test-case for the translator. Its files can be found in the following directory:

`/usr/people/spain/nets/I494`

#### **5.3.4.2 Program specifics**

The translator program files can be found in the following directory on the SGI workstation:

`/usr/people/spain/td_kronos`

This directory contains the entire set of TDFunction files, as well as the files added for the KRONOS translator. (From this point forward, a distinction will be made between original functions, TDFunctions, and the new KRONOS translation functions, TDKFunctions.) The newly added files are:

- kronos.h
- main.c
- kronos.c
- kinit.c

- ikronos.c
- translat.c
- all the KRONOS header files (in the directory k\_headers)

The code for these files, and for the makefile that is used to compile them, can be found in the appendix. The main program (main.c) functions as the translator by calling various TDFunctions and TDKFunctions.

The main program flow is summarized in figure 5.24 :

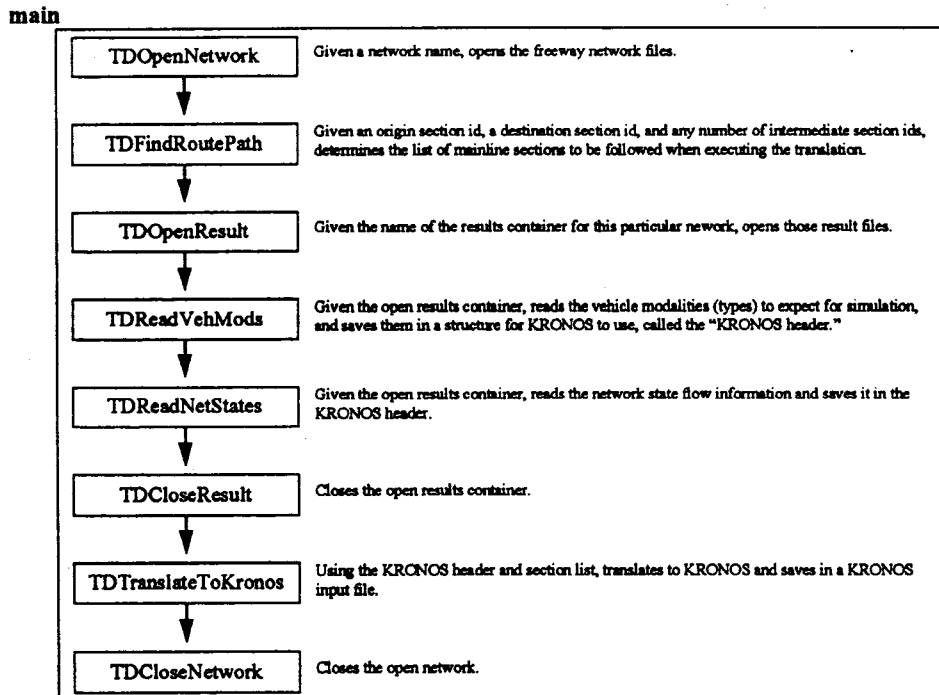


Figure 5.24 : Translator's Main Program Structure

Of special note are the TDFindRoutePath and the TDTranslateToKronos functions. TDFindRoutePath takes advantage of the fact that only a single freeway is being considered, and assumes that the user has supplied enough intermediate section ids to compute the mainline route. At each node, a survey is taken of the connected sections, and the section which is closest to heading in the direction of the next section in the list is chosen. This simple algorithm requires the correct intermediate sections to be indicated by

the user, or else it could wander down the wrong path and get stuck. Any large scale deployment of this translator would need a more general method.

TDTranslateToKronos is the heart of the translator. It takes the information gathered to that point, reads through the section list, and constructs the KRONOS segment types as data structures. These data structures are what are used to then create the KRONOS input file. The flow of this function is summarized in figure 5.25.

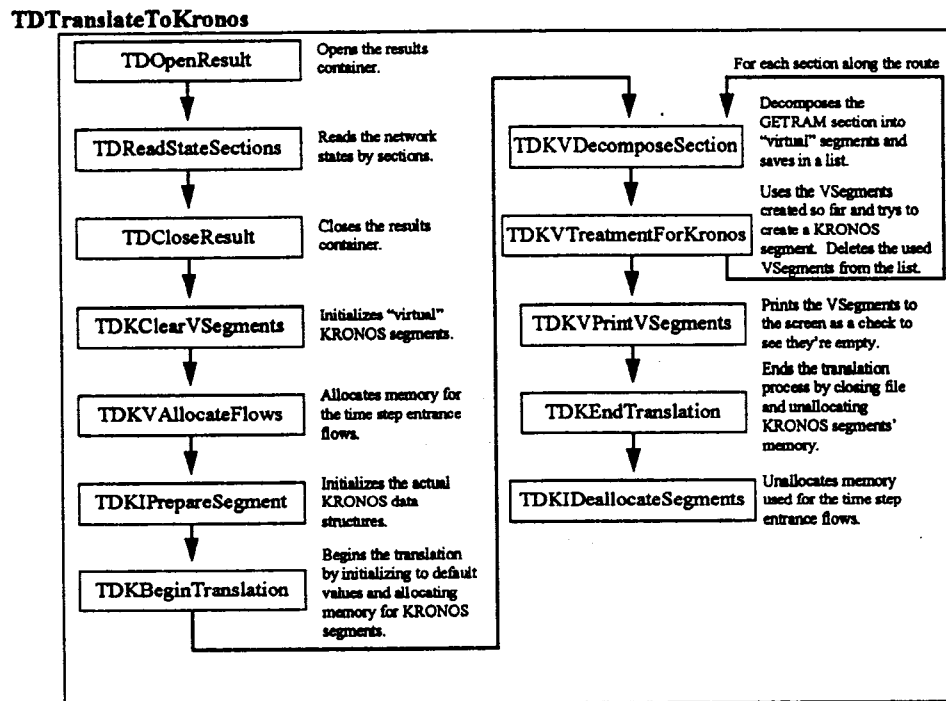


Figure 5.25 : TDTranslateToKronos Function Structure

### 5.3.4.3 Running the translator example

The translator example can be run inside a UNIX shell by first going to the directory /usr/people/spain/nets and typing "tdtest I494." The program will create the file I494.kr8, which can be transferred to a PC and run as a KRONOS (version 8) file.

## **6. IMPROVEMENT AND SUPPORT FOR THE MINNESOTA FREEWAY SIMULATION MODEL**

This chapter summarizes the improvements and support effort performed for this research for the Kronos freeway simulation model developed at the University of Minnesota. Most of the improvements were made to the MS-DOS based user interface and a series of workshops were conducted to show the applications of the software to the traffic engineers at the Minnesota Department of Transportation. The following sections summarize the revisions and the contents of the workshops.

### **6.1 Model Improvements**

#### ***Rearrangement of Windows***

There was no uniformity to the location and size of the buttons for each window within the previous interface. Thus, all of the windows within the input and output modules have been repositioned and reformatted for improved operations. The headers of the new windows also have a different color scheme. The locations of the buttons have been repositioned to maintain uniformity across a window. For example, after the modifications, the HELP button is located on left and OK button is located on right for each window.

#### ***Repositioning of data entry boxes***

The data entry boxes have been repositioned for each window to improve the appearance and clarity of each window.

#### ***Button Operation***

The way to activate a button within a window has been changed such that clicking the mouse on a button does not activate that button unless the mouse is released on the same button. This modification makes the module more consistent to the properties of push buttons within other windows-type software.



### ***Cursor's position***

The mouse cursor will remain in the same position where it was last placed, contrary to earlier versions.

### ***Naming of Variables***

Many variables within the module have been renamed to improve the readability and modification of the source code. This will save a substantial amount of time for implementing future enhancements.

### ***File Loading***

A new procedure has been added to the loading function which will issue error messages whenever an error in the data is found when loading an input data file. This procedure can help to identify the errors before running the simulation.

### ***Modifications and Bug Corrections in Geometry Data Input Windows***

- The simulation module requires the input module to convert the lengths of certain segments, such as the Simple OnRamp, Simple OffRamp and Weave sections, to the format recognizable by the simulator. A bug was causing the occasional disappearance of some segments from the geometric input screen. The code was reevaluated and the bug was fixed.
- For particular combinations of geometry, for example when the next to last segment of a freeway is an OnRamp, OffRamp or Weave section, it was not possible to exit the Edit Geometry Window by pressing the OK button. The only way of getting around this problem was to add another pipeline segment at the end of the freeway. This has been fixed.
- When a segment of the freeway geometry is highlighted using the mouse within the Edit Geometry, Initial Flow and Capacity, Demand, and Incident windows, the selected segment would be brought to the center of the freeway display associated with these windows. This feature was not well liked by the various users of the

module and was therefore was modified. Care has been taken to apply these changes to the Merge and Diverge Freeways also.

- The bug causing errors to the meter location within the OnRamp metering option was fixed.
- Earlier, the minimum number of lanes for the first segment of main freeway was limited to two lanes. To improve the flexibility of the module, it was decided to relax this restriction. Now a freeway can begin with a single traffic lane. In this case, extra care should be taken with incidents as KRONOS does not allow incidents to occur within one lane pipeline sections.
- In earlier versions, for the cases of lane add/drop segments, the number of lanes in “section C” were checked against a minimum or maximum lane limit. This resulted in an occasional error. It has been modified so that the check is done with the lengths in both Section A and Section B.
- An error within the Edit Geometry Window has been fixed and now the module displays the correct zone lengths and the correct number of segments.
- Due to the numerical problems within the traffic models, it was decided to increase the minimum length of a type 8 and a type 10 segment (LHS and RHS one lane On-Ramp) from 200 ft to 300 ft.

#### ***Bug Corrections in the Initial Flow and Capacity Window***

The previous input module did not check whether the initial flow on a segment was greater than the capacity. Since this error was not checked by the input module, the simulation module would stop executing such cases. This error checking is now performed in both the “Initial Flow and Capacity Window” and the “Demand Window”. Data which does not satisfy this condition is detected as soon as a violation takes place in either window.

### ***Bug Corrections in Incident Window***

KRONOS requires a pipeline section to be at least 800 ft in length in order to place an incident. The program checks whether there is any such segments within the given freeway geometry before calling the incident window. If there are no such segments, the program will display an error message. This check, however, did not include the merging and diverging freeways. The program would not allow user to place an incident if the main freeway did not have any 800 ft+ segment even though the merging or diverging freeways had such segments. This has been fixed.

### ***Bug Corrections in the Factor Button of the Demand Window***

- The demand factor option of the Demand Window would set the flows to capacity if the applied factor caused the resulting flow to exceed the segment capacity. The modified version asks the user for confirmation before changing those demand values.
- In the previous version, the demand factor button would retain the previously applied value. It is now set to a default value of 1.0 each time that window is called.
- If the user enters a value of zero as a factor, the module prompts for confirmation to alert the user of a possible mistake before applying that factor.
- The problem that caused the factor button to not work for the on-ramp with an exclusive lane addition and the off-ramp with a lane drop has been fixed.

### ***Modification to Start Simulation Option***

A new error message has been added if any attempt to start a simulation run before a file is loaded or a new file is created occurs. Further, the routine for the Help button has been examined and the bugs that caused the malfunction of the restoring windows were fixed.

### ***Modification of Goto DOS Shell Option***

The “go to DOS shell” option was modified so that when within the DOS shell, the user can do DOS application provided there is enough memory.

### ***Modifications to Exit Option***

An additional window has been added to give the user an opportunity to abort the exit action.

### ***Summary Window***

The summary window would not show the correct file name when there was a problem loading a file. This problem has been fixed and the data in the summary window has been rearranged for improved legibility..

### ***Enhancement of Error Checking Function***

The error checking functions for each data value have been enhanced to keep the input data from falling outside of any allowable ranges.

### ***Addition of Print MOE Table option to the Output module***

A new print function was added to the output module to allow for the printing of a selected portion of the MOE table. Using this function, a user can print the values of the measure of effectiveness for certain time periods. In previous versions, the user had to print the MOE table for the entire simulation.

## **6.2 Support for Model Implementation**

A series of workshops have been conducted for the traffic engineers of the Minnesota Department of Transportation to help them learn about the Kronos software. A total of four workshop sessions were offered at the ITS Institute at the University of Minnesota.

The specific dates and issues for each session are as follows;

- Session 1: January 12, 1996, 9:00 - 11:30 a.m., ITS Lab  
Topic: Issues in Freeway Operations

- Session 2: January 26, 1996, 9:00 - 11:30 a.m., ITS Lab  
Topic: Modeling Approaches in Kronos
- Session 3: February 2, 1996, 9:00 - 11:30 a.m., ITS Lab.  
Topic: Hands-on Experiments with Kronos 8
- Session 4: February 9, 1996, 9:00-11:30a.m., ITS Lab.  
Topic: New Features in the Kronos 9 and Future Directions

### **6.2.1 Session One: Issues on Freeway Operations**

The following issues were identified as ongoing problems in freeway operations :

- Capacity.
- Bottleneck.
- Optimal Balancing of Flows.
- Congestion.
- Travel time / Delay (variability).
- Level of Service.
- Weaving.
- Quantifying Ramp Meter Benefits.
- Peak Period.
- Weather.
- Incidents.
- Motorist Information.

Specific discussion of the major issues is summarized as follows:

- The existing traffic models do not effectively consider the interaction between human and vehicle elements.
- Capacity has not been specifically defined for traffic operations. Depending upon the objectives, the capacity could be considered as a constant value for a given geometric condition or a variable affected by the traffic conditions.
- No proper definition for the capacity at a weaving section on a freeway exists. The length of a weaving section could extended to as long as 3500 ft. In the type A weaving section, it may be possible to increase the capacity of a weaving section by metering the on-ramp input volume.

- Since the maximum volume measured over different time intervals can vary significantly, the definition of capacity should also involve time intervals.
- Occupancy measurements can change depending on the calibration factor of each loop detector station.
- Currently the following formula is being used to estimate speed from the volume and occupancy measurements.

$$Speed = \alpha \left( \frac{5 \text{ min Volume}}{5 \text{ min Occupancy}} \right)$$

The value of  $\alpha$  differs very much with different sections. The effect of different time periods (e.g. 10 min., 15 min., 30 min.) on the value of  $\alpha$  needs to be studied.

- Although it is usually thought that the lane capacity of a freeway is about 2300 vph, field observations indicate that flows as high as 2800 vph can exist when the capacity is measured in 5 minute increments.
- When congestion is released and the drivers begin to accelerate, the headways tend to be longer than the deceleration headways. This results in a lower maximum volume when compared to the value before the congestion happened. Future research needs to study how the lower maximum volume should be interpreted and used for traffic operations.

Other issues raised by the participants during the subsequent discussion include :

- Quantification of the benefits of ramp metering .
- Measurement of ramp waiting delay.
- Estimation of diversion traffic because of ramp queue.
- Installation of variable signs indicating ramp waiting delay.

### 6.2.2 Session 2: Traffic Modeling in Kronos

The objective of this session was to introduce the concepts of modeling adopted in KRONOS to the practicing engineers. The session started with a general introduction

of major modeling approaches used in practice. E.g. : Microscopic, Macroscopic and Mesoscopic modeling approaches.

#### **6.2.2.1 Major features in Kronos Modeling**

KRONOS adopts the simple continuum modeling approach and tries to achieve flow conservation and capture the propagation of shock waves by using numerical schemes developed by Lax. The main parameter of the model is the QK relationship which is determined by three coordinates in the q-k plane.

##### *Treatment of Capacity Changes*

Often we see changes in capacity along the freeway. These changes are handled in KRONOS by applying a capacity gradient across the boundary where the change is observed. The discussion focused on what the shape of the gradient to be used should be. The current modeling methodology of KRONOS uses a straight line variation which is restricted to 2 DXs.

##### *Upstream and Downstream Boundary Treatment*

KRONOS allows two different flow conditions at the upstream boundary.

- Option 1 : Only the entry flow is given. A dummy DX is placed at the upstream boundary for modeling purposes. This DX can have density. This option is useful in modeling hypothetical demand conditions.
- Option 2 : entry flow and congestion condition specified. The congestion condition indicates whether the entry flow is congested or not. This option can be used when the user wants to impose a certain flow conditions on an incoming arrival flow at the upstream boundary or to calibrate the model with a real data set.

There are three options to specify the downstream boundary flow conditions;

- Option 1 : No specification. One dummy DX at the downstream end will be generated and the density of this DX is assumed to be similar to the last DX. This is the most commonly used option.
- Option 2 : Only the exit demand is specified. In this case, the user specifies an exit flow which will be extracted from the last dx of the given freeway. This option is often used to analyze hypothetical situations.
- Option 3 : Both exit demand and the downstream congestion conditions are specified. In this case, the flow and density of the last dx of a given freeway is set to the same values specified by the user. The density values are estimated from the q-k relationship with the flow and congestion conditions entered by the user. This option is recommended when a user wants to impose a certain set of exit conditions or to calibrate a model with a real data set.

#### *Merging Modeling at Entrance Ramp Areas*

In the current model, the number of vehicles entering the main freeway from an on-ramp is assumed to vary parabolically as a function of the available space on the main freeway. It was pointed out that the capacity of the on-ramp section varies according to the length of the acceleration lane. Shorter acceleration lanes may reduce the capacity as there is higher probability for the ramp traffic to yield to the mainline traffic. The effect of different on-ramp capacities for the merging flow needs to be considered in the future enhancement of the merging model. Ramp capacities vary from 1700 (when there is no lane addition) to 2400 (when there is a lane addition).

#### *Diverging Model at Exit Ramps*

The current diverging model requires the user to specify an additional diversion rate through time. The issues regarding the estimation of a diverting flow and the treatment of spillback from an off-ramp section were discussed and further research in this area was recommended. Also, the lane friction effects, the reduction of speed at adjacent lanes when the right-most lane is congested, should be accounted for in future modeling.



### *Weaving Model*

The modeling issues identified for weaving areas include;

- The quantification of a weaving conflict and the effects of merging/diverging flows in the weaving section on the total throughput.
- The effects of the off-ramp capacity on the diverging flow. The speed of diverging flows can vary significantly even without congestion.
- The effects of a diverging flow on a merging flow.
- The lane friction effect due to the right-most lane diverging traffic on the through capacity.

### **6.2.3 Session 3: Hands-on Experience with Kronos 8**

In this session, the KRONOS 8.0 software was demonstrated to the participants and each participant was given an opportunity to try the software. The participants were divided into several groups of two and were asked to create hypothetical analysis cases using real geometry data from the Twin Cities' metro freeways. The participants also analyzed the output from each simulation using the graphical output module.

### **6.2.4 Session 4: New Features in Kronos 9**

The new features to be added in the next version of Kronos were presented in this session. These are,

- Complete migration to Windows operating environment.
- Bi-directional network simulation.
- HOV Lane Module.
- Rate-selection ramp metering module.
- Automatic Calibration of Q-K Curve.

The use of Kronos as an operational decision support tool was also discussed as the major direction for the future development.

## 7. CONCLUSIONS AND FUTURE RESEARCH NEEDS

This study evaluated six widely available freeway simulation models using a common data set. A systematic testing plan was developed with real and hypothetical data to evaluate the quantitative as well as qualitative behavior of each model. A total of three quantitative cases were developed with the real traffic data collected from the Twin Cities freeway network, while six qualitative cases were developed using hypothetical data to evaluate the conservation and spillback characteristics of each model.

Several problems have been identified in testing simulation models. First, there is a lack of real data for diverse geometric and operational conditions. For example, the weaving area of cloverleaf intersections usually did not have detectors. Further, the freeway detectors located within the I-94 eastbound construction zone were not operational. In addition, vehicle classifications can not be determined from the detector data and need to be collected manually. Also, the current configuration of the Twin Cities detectors does not allow the measurement of speed. Secondly, there is a need for an efficient procedure to check the validity of the real data. Finally, there is a strong need to develop an automatic calibration procedure to determine the best set of the parameters for each model for a given geometric and traffic conditions. In this research, the time consuming trial-and-error method was used to calibrate each model, which makes the fair comparison of each model very difficult.

In addition to evaluating the freeway models, a basic framework for an automatic simulation environment was also developed in this research. The main objective of the automatic simulation environment is to automate the time consuming input data preparation process, so that traffic engineers can spend more time in analyzing data rather than preparing input files for simulation. A prototype system using the GETRAM software was also developed and the basic concept of automatic simulation was demonstrated.

Future research needs include the development of a comprehensive data sets with various geometric and traffic conditions including speed measurements, an efficient

procedure to test the quality of the data set, and an automatic model calibration system that can determine the best set of the parameters for each model for a given geometric and traffic conditions. Further, the automatic simulation system where data processing, input file preparation and analysis of the data can be efficiently performed in an integrated environment needs to be developed.

## References

1. FRESIM User Guide, Version 4.5. Federal Highway Administration, Wash. D.C., April 1994.
2. "Tutorial", Fresim Computer Simulation Package. Federal Highway Administration, November 1994.
3. AIMSUN2 User's Manual. Universitat Politecnica de Catalunya, Barcelona, Spain, December 1995.
4. Tedi: The Getram Editor User's Guide. Universitat Politecnica de Catalunya, Barcelona, Spain, December 1995.
5. Barcelo, J., J.L. Ferrer and R. Grau. "AIMSUN2 and the GETRAM Simulation Environment", Universitat Politecnica de Catalunya, Barcelona, Spain, 1994.
6. Barcelo, J., J.L. Ferrer and R. Grau. "Microscopic Traffic Simulation for ATT Systems Analysis", Universitat Politecnica de Catalunya, Barcelona, Spain, 1996.
7. Van Aerde, M. INTEGRATION : A Model for Simulating IVHS in Integrated Traffic Networks, User's Guide for Model Version 1.5e. Transportation Systems Research Group, Queens University, Kingston, Ontario, Canada, August 1994.
8. Van Aerde, M. INTEGRATION: An Overview of Traffic Simulation Features. Department of Civil Engineering, Queens University, Kingston, Ontario, Canada, 1996.
9. Van Aerde, M. QueensOD User's Guide for Model Version 1.2c. Transportation Systems Research Group, Queens University, Kingston, Ontario, Canada, August 1994.
10. Van Aerde, M. and Mark Baker. Weaving Capacity Sensitivity Analysis Using the INTEGRATION Model. Queens University, Kingston, Ontario, Canada, 1996.
11. TRAF User Reference Guide, Publication No. FWHA-RD-92-060. Federal Highway Administration, Wash. D.C., May 1992.
12. Payne, Harold J. "FREFLO: A Macroscopic Simulation Model of Freeway Traffic". Transportation Research Record 722, 1979, pp.68-77.

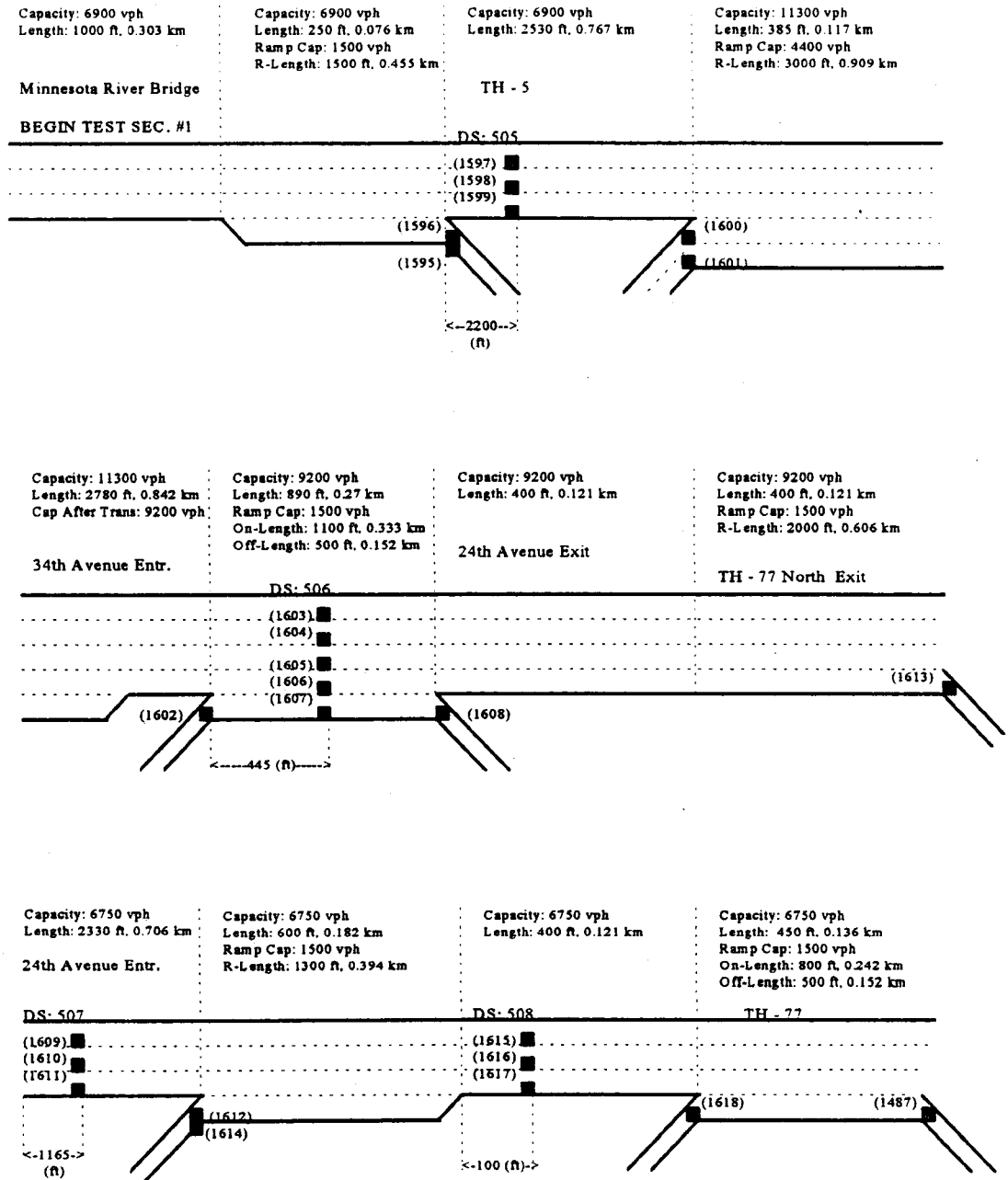
13. Imada, Tsutomu and Adolf A. May. FREQ8PE: A Freeway Corridor Simulation and Ramp Metering Optimization Model. Institute of Transportation Studies, University of California, Berkeley. June 1985.
14. KRONOS User's Guide. University of Minnesota, Minneapolis, Minnesota, December 1994.
15. Michalopoulos, P. and Eil Kwon. "Macroscopic Simulation of Traffic Flows in Complex Freeway Segments on a Personal Computer", University of Minnesota, 1995.
16. McShane, William R. and Roger P. Roess. Traffic Engineering, Prentice Hall, Englewood Cliffs, New Jersey, 1990.

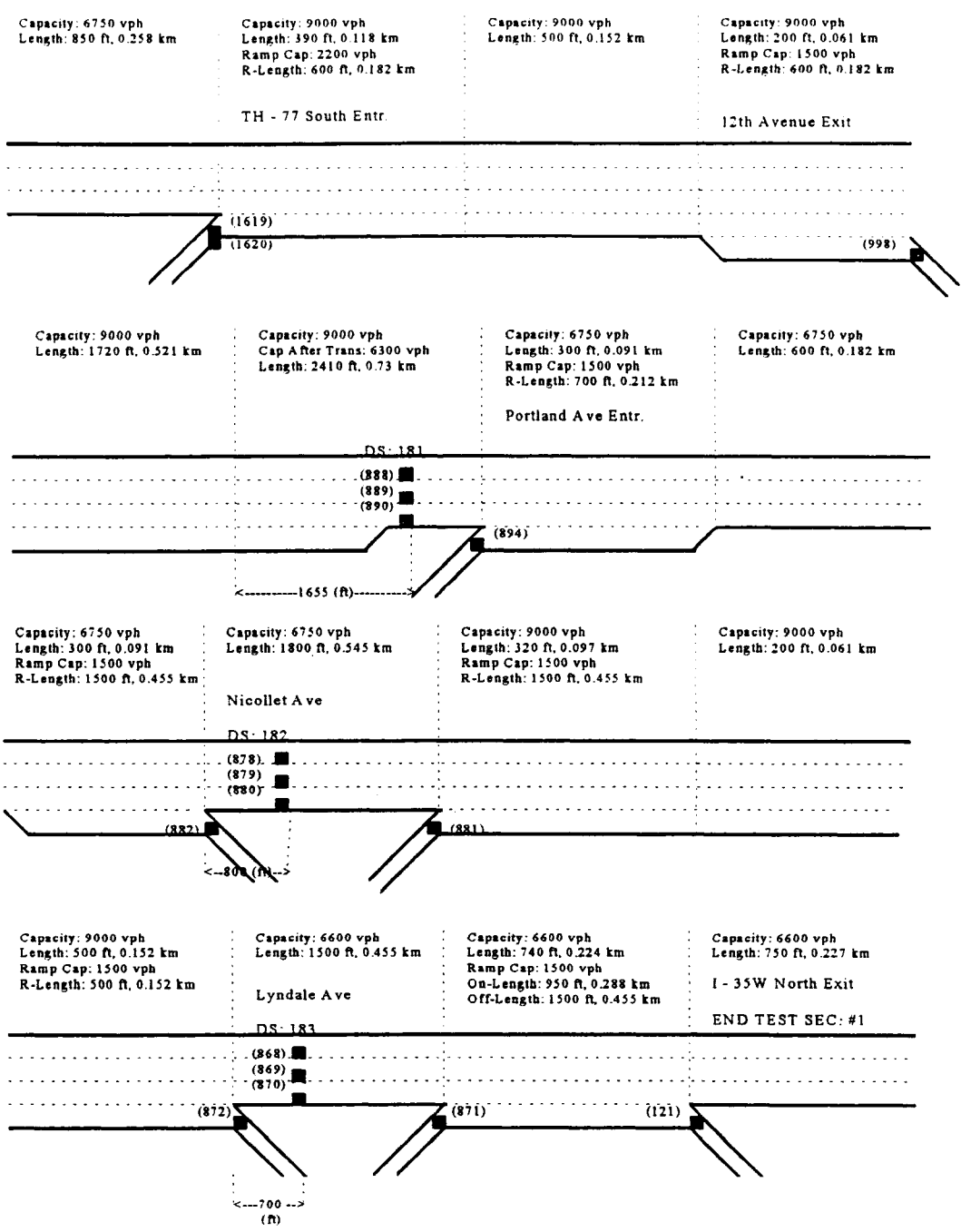
## Appendix A

### Quantitative Test Cases

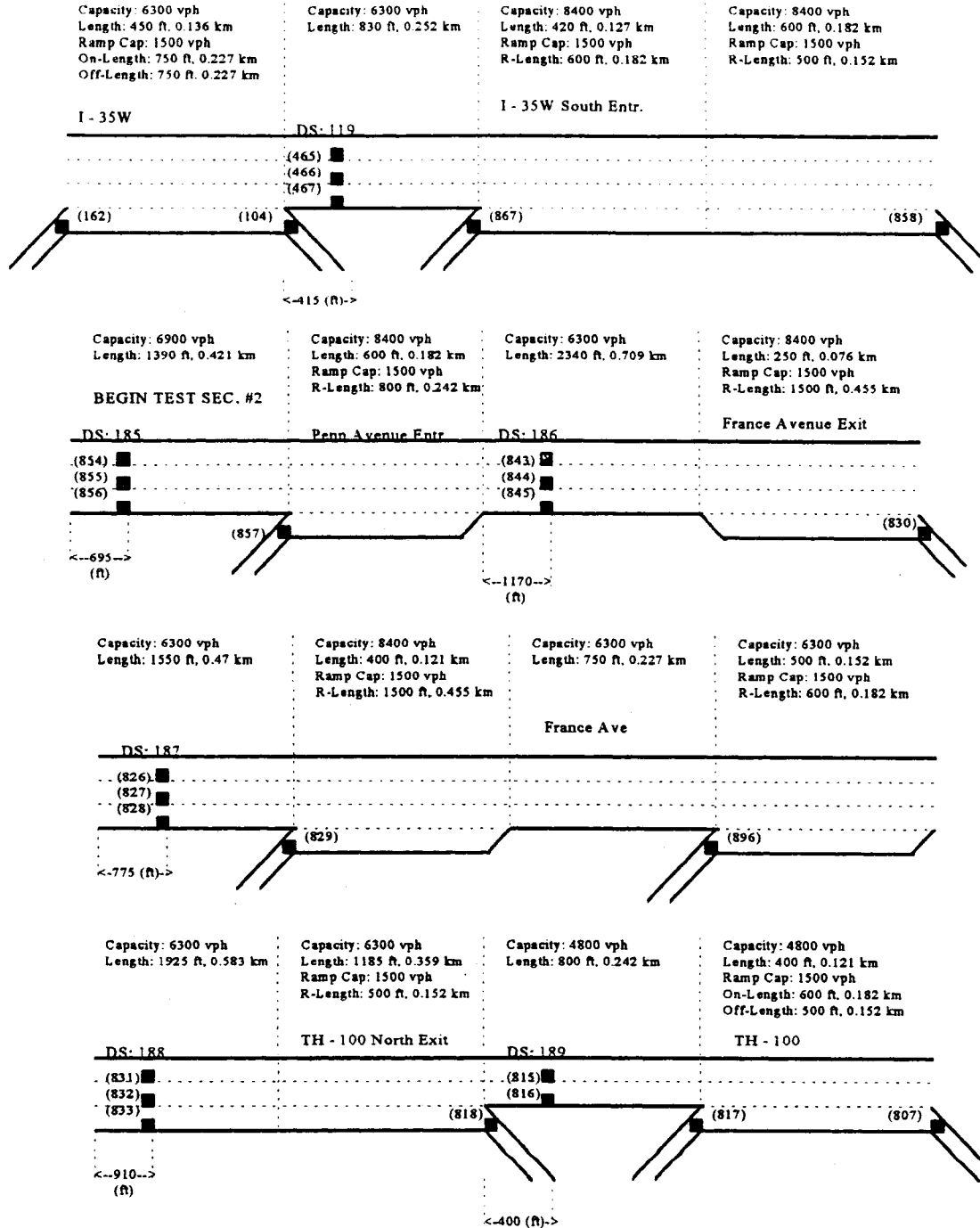
I-494 Westbound Geometrics	A-2
I-494 Westbound Mainline Volume and Occupancy Data	A-8
I-494 Westbound Ramp Volume and Occupancy Data	A-25
Calibrations for I-494 WB South of the Twin Cities International Airport	A-34
Calibrations for I-494 WB Through Southwest Minneapolis	A-36
I-94 Eastbound Geometrics	A-38
I-94 Eastbound Mainline Volume and Occupancy Data	A-42
I-94 Eastbound Ramp Volume and Occupancy Data	A-55
Calibrations for I-94 EB During the University of Minnesota Ramp Construction	A-59

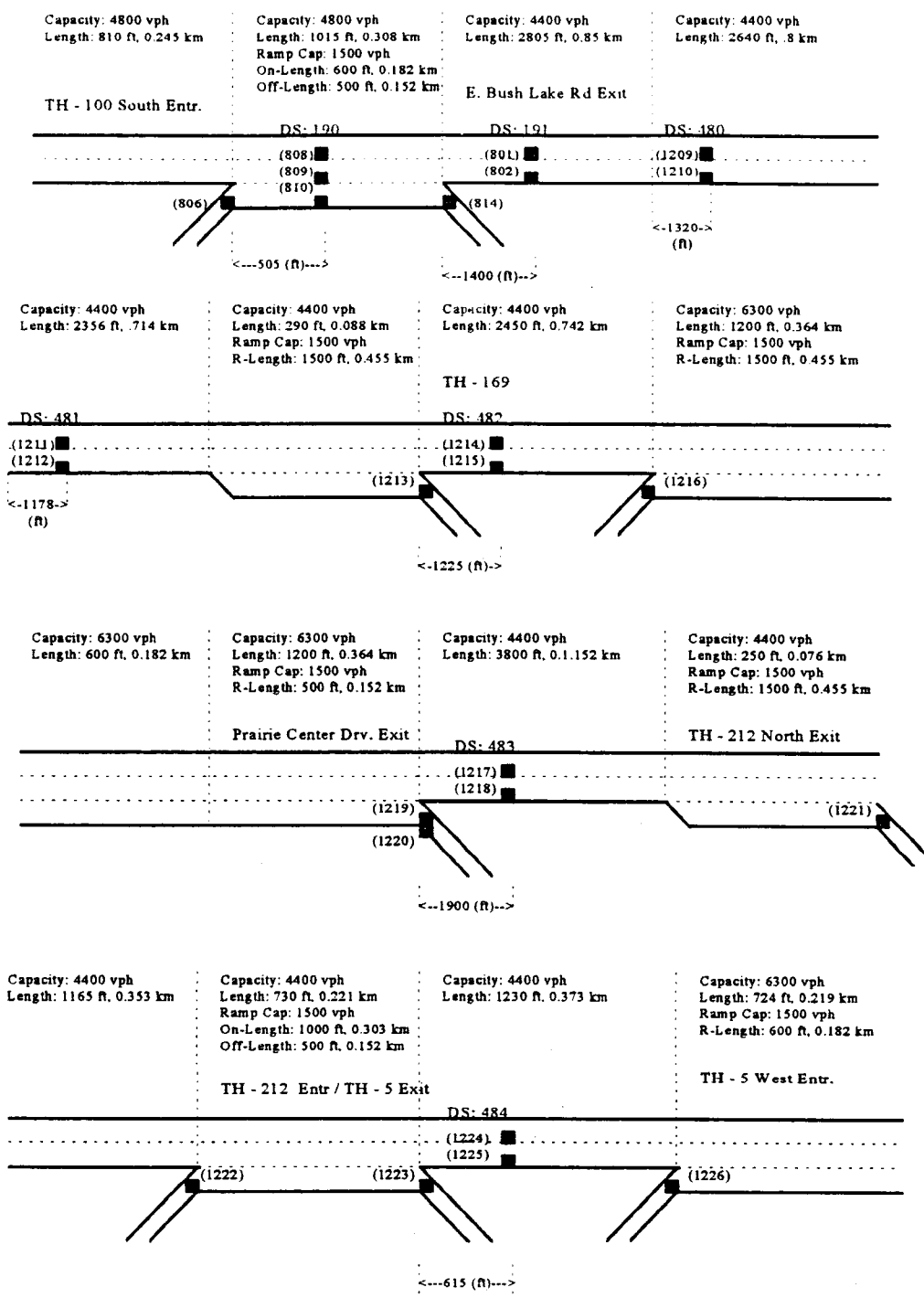
## TEST FREEWAY SECTION (I-94 Westbound, Minneapolis)

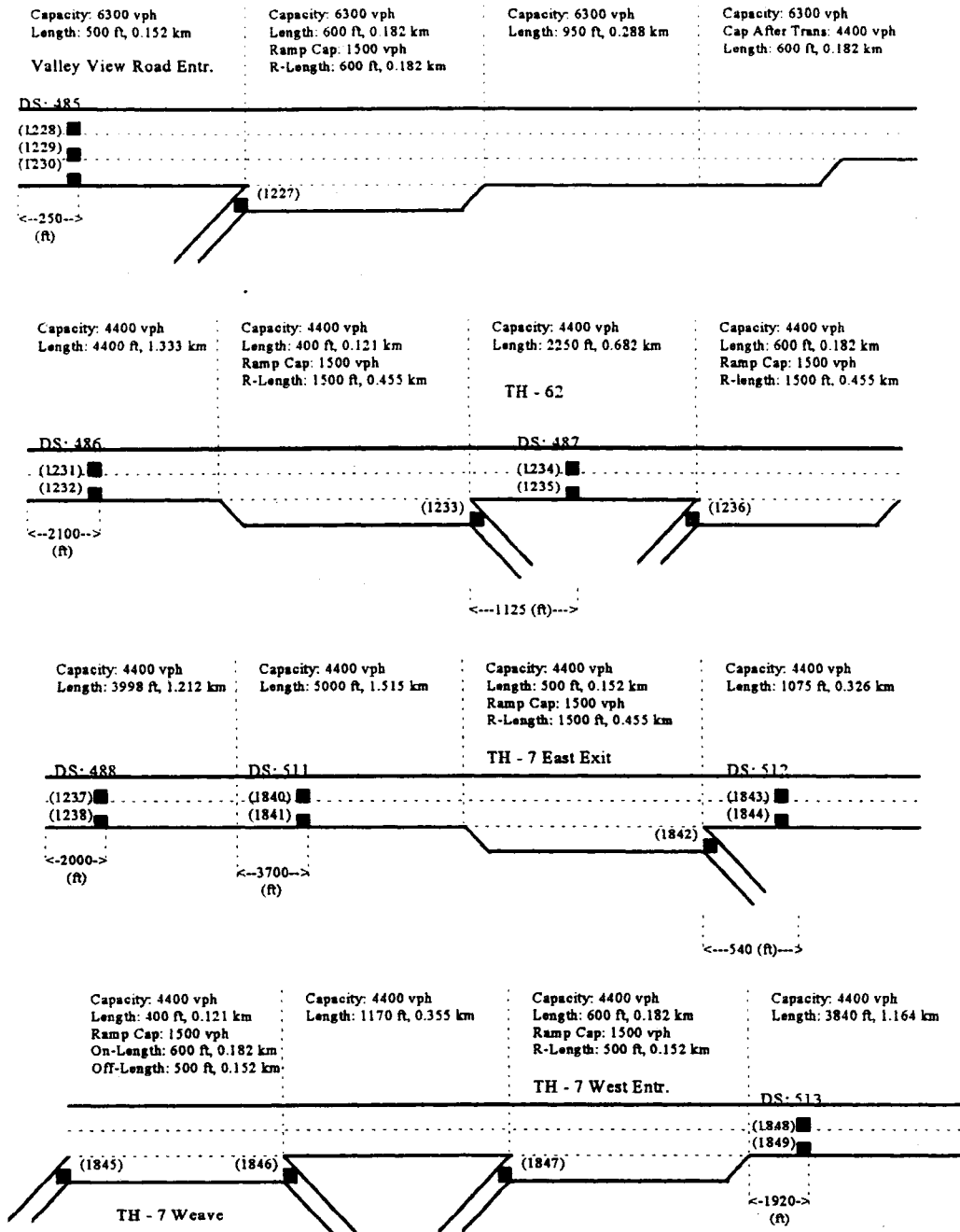


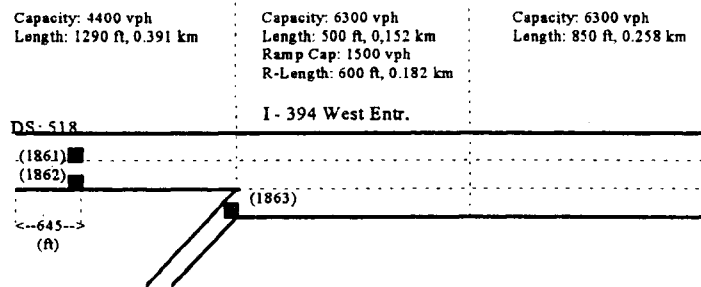
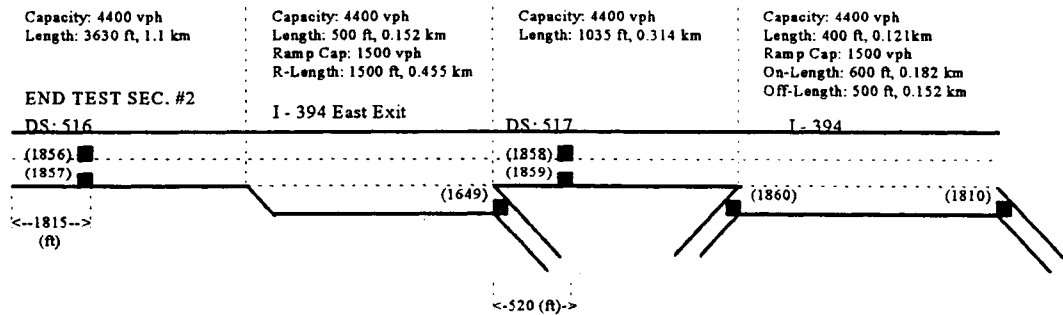
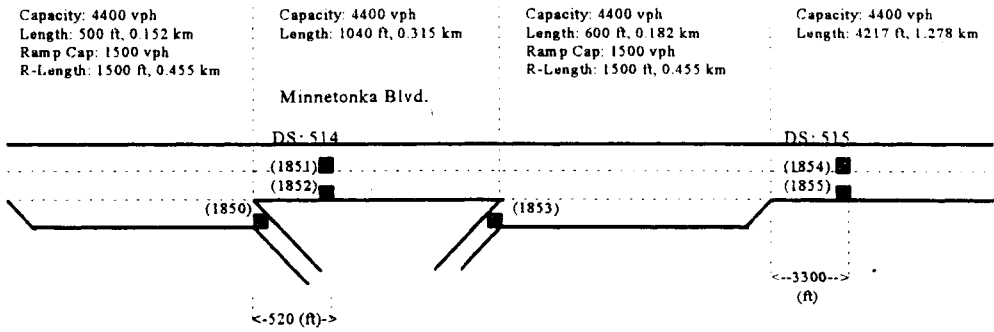












DETECTOR STATION 505

TIME	detector 1597		detector 1598		detector 1599	
	VOL	OCC	VOL	OCC	VOL	OCC
7:00	696	5	1068	7	1680	9
7:05	732	5	1068	6	1584	9
7:10	708	5	1140	7	1428	7
7:15	672	5	1188	7	1776	9
7:20	768	6	1260	8	1932	11
7:25	936	6	1356	9	2268	12
7:30	996	6	1440	10	2268	12
7:35	1200	8	1608	10	2112	12
7:40	840	5	1392	8	1992	11
7:45	984	6	1260	7	1968	11
7:50	1188	8	1308	9	2280	13
7:55	1068	6	1380	8	1788	10
8:00	1032	6	1344	8	1692	9
8:05	828	5	1092	7	1440	8
8:10	744	4	1056	7	1644	9
8:15	672	4	972	6	1248	6
8:20	672	4	1224	7	1608	8
8:25	720	4	1056	8	1536	8
8:30	624	4	1236	9	1476	7
8:35	720	5	1032	6	1068	6
8:40	588	4	876	6	1092	6
8:45	636	5	1044	8	1176	6
8:50	636	4	1176	7	948	5
8:55	624	5	924	6	900	5
9:00	528	4	1008	7	1044	6
9:05	492	4	840	6	804	4
9:10	564	4	876	6	684	4
9:15	396	3	888	6	684	3
9:20	468	3	804	7	648	3
9:25	516	4	720	4	708	4
9:30	456	3	816	5	456	2
9:35	504	4	708	5	564	3
9:40	540	4	780	7	816	5
9:45	480	3	924	6	684	3
9:50	456	3	816	5	528	2
9:55	504	4	732	6	576	3
10:00	588	4	852	7	708	4

DETECTOR STATION 506

TIME	detector 1603		detector 1604		detector 1605	
	VOL	OCC	VOL	OCC	VOL	OCC
7:00	636	4	900	6	936	6
7:05	624	5	504	3	1272	8
7:10	432	3	624	4	1152	7
7:15	396	3	756	5	1236	9
7:20	420	2	624	3	1260	9
7:25	816	6	576	4	1476	10
7:30	552	3	480	3	1524	10
7:35	864	6	468	2	1620	11
7:40	552	4	516	4	1440	9
7:45	672	5	552	3	1356	9
7:50	900	6	636	4	1584	11
7:55	1080	8	660	4	1572	10
8:00	1104	8	636	4	1500	10
8:05	1032	7	576	4	1044	7
8:10	900	7	528	3	1008	6
8:15	684	4	456	2	1080	7
8:20	708	5	444	3	1092	7
8:25	852	6	276	1	1032	7
8:30	564	4	684	4	1068	7
8:35	648	4	588	4	1128	7
8:40	660	5	492	3	1008	6
8:45	660	5	672	5	1080	8
8:50	660	4	696	5	888	5
8:55	588	4	552	4	1104	8
9:00	672	5	504	3	1044	7
9:05	672	4	660	4	840	6
9:10	600	4	636	4	1020	7
9:15	444	3	612	4	768	5
9:20	372	2	576	4	780	6
9:25	600	4	540	3	684	4
9:30	468	3	600	4	756	5
9:35	528	3	516	3	720	5
9:40	456	3	624	4	828	6
9:45	492	3	696	4	816	5
9:50	504	3	588	3	720	4
9:55	408	3	468	3	768	5
10:00	708	5	480	3	732	5

DETECTOR STATION 506

TIME	detector 1606		detector 1607	
	VOL	OCC	VOL	OCC
7:00	1380	9	1704	10
7:05	1188	8	1776	10
7:10	1248	8	1560	9
7:15	1344	8	1668	10
7:20	1584	11	2004	12
7:25	1764	12	2196	13
7:30	1644	12	2184	13
7:35	1728	11	2208	14
7:40	1596	10	1980	12
7:45	1632	10	1980	12
7:50	1560	10	1980	13
7:55	1572	10	1776	11
8:00	1416	10	1620	10
8:05	1308	9	1404	8
8:10	1188	8	1488	9
8:15	1152	7	1248	7
8:20	1296	8	1464	8
8:25	1248	9	1716	10
8:30	1428	10	1284	7
8:35	1068	7	1164	6
8:40	1140	8	1080	6
8:45	1164	9	1320	8
8:50	1140	7	1140	6
8:55	1176	8	1044	6
9:00	1092	8	1176	7
9:05	1008	6	816	5
9:10	900	6	624	4
9:15	900	6	600	3
9:20	948	9	648	4
9:25	792	5	612	3
9:30	900	6	480	2
9:35	732	5	612	3
9:40	840	7	744	4
9:45	852	5	672	3
9:50	756	5	636	3
9:55	792	6	588	3
10:00	888	7	576	3

DETECTOR STATION 507

detector 1609		detector 1610		detector 1611	
VOL	OCC	VOL	OCC	VOL	OCC
1512	11	1404	10	1812	11
1512	10	1296	9	1812	11
1428	10	1272	9	1560	9
1440	10	1476	10	1596	9
1404	10	1560	11	2088	12
1728	13	1656	13	2172	14
1944	13	1704	13	2196	14
1836	13	1812	13	2064	13
1800	12	1644	12	1980	12
1560	11	1644	12	1968	13
2064	16	1584	13	1872	13
1836	16	1548	13	1800	13
1572	32	1464	25	1392	30
1668	17	1356	11	1548	10
1404	9	1128	7	1440	8
1308	8	1140	8	1368	7
1380	9	1248	8	1404	8
1272	8	1320	9	1704	9
1308	9	1368	9	1512	8
1236	8	1260	9	1176	6
1272	8	1116	8	1080	6
1128	8	1416	12	1332	8
1104	7	1140	7	1200	7
1260	10	1188	9	1200	7
1392	10	1116	9	972	6
1104	9	1044	7	924	5
1284	10	1044	7	648	4
1080	8	864	6	492	2
1032	8	1008	9	792	4
840	6	996	7	540	3
996	7	888	6	540	3
744	5	852	6	636	4
960	7	852	7	720	4
984	7	996	7	660	3
900	5	924	6	624	3
960	7	840	6	600	3
876	6	876	7	528	3

DETECTOR STATION 508

TIME	detector 1615		detector 1616		detector 1617	
	VOL	OCC	VOL	OCC	VOL	OCC
7:00	1680	12	1344	9	1896	11
7:05	1248	8	1416	9	1620	9
7:10	1644	12	1440	10	1584	9
7:15	1476	11	1776	13	2088	12
7:20	1692	14	1728	13	2220	14
7:25	2028	14	1728	13	2184	13
7:30	1944	15	1788	14	2112	15
7:35	1884	16	1752	16	1956	16
7:40	1824	16	1620	16	1992	22
7:45	2028	18	1668	14	1836	14
7:50	1920	27	1536	19	1776	18
7:55	1680	27	1488	35	1296	30
8:00	1764	17	1608	16	1572	17
8:05	1572	11	1284	8	1524	9
8:10	1332	9	1224	9	1440	7
8:15	1368	10	1392	9	1488	8
8:20	1284	8	1452	10	1704	9
8:25	1404	9	1416	9	1572	8
8:30	1236	8	1320	8	1152	6
8:35	1284	8	1284	9	1200	7
8:40	1356	10	1548	11	1380	8
8:45	1356	9	1332	9	1284	7
8:50	1368	11	1308	10	1272	7
8:55	1560	11	1116	8	1080	6
9:00	1296	10	1272	9	984	6
9:05	1224	9	1236	8	684	4
9:10	1272	9	1092	8	528	2
9:15	1068	9	1128	9	852	4
9:20	888	6	1092	8	576	3
9:25	1104	8	1044	6	612	3
9:30	1020	7	960	7	576	3
9:35	1020	8	960	8	828	5
9:40	1128	8	1164	8	768	4
9:45	1044	7	1032	6	660	3
9:50	1128	9	1008	7	684	3
9:55	1092	8	1032	8	516	3
10:00	1068	7	840	6	444	2

DETECTOR STATION 181

detector 888		detector 889		detector 890	
VOL	OCC	VOL	OCC	VOL	OCC
1572	9	1800	10	2016	9
1668	10	1920	10	2316	11
1404	7	1764	9	1932	9
1500	8	1884	10	1884	9
1668	9	2124	12	2304	12
1884	13	1956	13	2448	15
2064	24	1740	18	2136	18
2028	18	1956	16	2160	15
1836	19	1728	18	1884	21
1956	20	1932	19	2124	19
1884	24	1656	20	1824	24
1884	19	1752	23	1836	18
1884	19	1752	23	1836	18
1740	28	1536	23	1668	26
1752	19	1728	18	1884	19
1752	17	1548	15	1572	17
1608	9	1680	10	1896	9
1416	8	1644	9	1812	9
1464	7	1740	10	1944	9
1416	7	1644	9	1752	8
1404	7	1500	8	1524	7
1512	8	1872	11	1848	8
1668	10	1812	12	1956	11
1536	9	1740	10	1704	8
1368	8	1428	8	1416	7
1308	8	1476	8	1344	6
1092	6	1392	7	1032	5
1188	6	1248	6	780	3
1296	7	1272	8	1056	5
1236	7	1236	7	888	4
1152	6	1416	7	780	3
1224	6	1224	6	684	3
1212	6	1236	7	972	4
1368	7	1380	8	1104	5
1236	6	1356	7	804	3
1236	6	1164	6	972	4
1308	6	1308	7	816	3

DETECTOR STATION 182

TIME	detector 878		detector 879		detector 880	
	VOL	OCC	VOL	OCC	VOL	OCC
7:00	1848	6	1608	8	2160	15
7:05	1968	7	1212	6	2280	14
7:10	1884	7	1152	5	2076	12
7:15	1800	6	1356	6	2076	12
7:20	1968	7	1488	7	2328	15
7:25	1992	7	1524	9	2280	16
7:30	1920	7	1908	11	2232	15
7:35	1860	6	1776	11	2040	22
7:40	1884	7	1704	10	2088	16
7:45	1812	6	1656	13	1968	29
7:50	1344	5	1680	18	1380	40
7:55	1728	6	1836	13	1788	28
8:00	1776	6	1740	13	1896	23
8:05	1476	5	1584	18	1560	39
8:10	1716	6	1644	14	1860	25
8:15	1680	6	1644	12	1776	25
8:20	1632	6	1536	11	1668	33
8:25	1512	5	1692	11	1788	22
8:30	1776	6	1440	7	1920	13
8:35	1572	5	1356	6	1956	11
8:40	1452	5	1200	5	1476	8
8:45	1680	6	1320	6	1968	11
8:50	1836	6	1644	9	2208	15
8:55	1668	6	1524	9	1824	20
9:00	1428	5	1464	14	1524	17
9:05	1620	6	1236	6	1596	10
9:10	1260	4	1116	5	1068	6
9:15	1248	4	1044	4	924	5
9:20	1260	4	1212	5	1176	7
9:25	1212	4	1080	5	972	5
9:30	1392	5	936	4	1116	6
9:35	1344	5	996	4	720	4
9:40	1332	5	1032	5	1104	6
9:45	1488	5	1332	6	1164	7
9:50	1284	4	1200	5	936	4
9:55	1284	4	1116	5	984	5
10:00	1248	4	1128	5	948	5

DETECTOR STATION 183

detector 868		detector 869		detector 870	
VOL	OCC	VOL	OCC	VOL	OCC
1644	13	2064	18	2220	18
1212	10	1944	16	2160	17
1200	9	1740	14	2208	16
1332	9	1920	16	1992	14
1416	12	1776	16	2220	18
1488	14	1944	23	2268	24
1512	17	1968	27	2268	32
1656	17	1776	24	2004	28
1692	20	1764	29	1992	31
1464	18	1776	36	1920	31
1608	24	1296	44	1536	39
1764	23	1668	35	1824	36
1524	25	1428	44	1608	42
1608	27	1620	33	1884	33
1596	26	1572	41	1896	35
1440	32	1716	38	1740	36
1596	22	1560	37	1764	32
1644	20	1608	37	1920	33
1584	22	1620	34	2004	34
1320	14	1740	24	1968	25
1116	8	1560	12	1500	11
1236	9	1752	14	1908	13
1620	16	1800	21	2160	25
1356	22	1296	40	1416	42
1500	27	1452	33	1548	28
1548	19	1668	22	1752	20
984	7	1176	8	1044	8
1272	9	1236	9	1080	7
1272	9	1248	10	1020	7
1332	10	1200	10	1152	8
1176	9	1332	9	1056	7
1032	7	1488	11	900	6
1260	10	1428	12	1068	7
1356	11	1416	12	1236	8
1236	8	1356	10	900	5
1164	9	1272	9	1068	6
1188	9	1212	10	936	5



DETECTOR STATION 119

TIME	detector 465		detector 466		detector 467	
	VOL	OCC	VOL	OCC	VOL	OCC
7:00	2100	19	2004	14	2388	15
7:05	2040	20	2076	17	2268	18
7:10	2004	18	2112	15	2244	14
7:15	1860	16	2040	13	1968	11
7:20	1944	19	1932	14	2364	15
7:25	2196	21	2004	20	2304	18
7:30	2016	26	2004	20	2316	21
7:35	1956	21	2052	20	2184	16
7:40	2064	24	2112	20	2184	17
7:45	1740	26	1800	29	1836	25
7:50	1824	28	2052	30	2052	22
7:55	1812	33	1956	23	2076	20
8:00	1524	31	1704	31	1812	29
8:05	1824	31	1956	24	2028	19
8:10	1644	31	1920	26	2196	21
8:15	1656	33	1980	25	2124	24
8:20	1644	34	1656	33	1968	24
8:25	1764	23	1956	22	2100	19
8:30	1872	27	1884	26	2196	21
8:35	1812	28	1884	23	2148	17
8:40	1800	24	1920	19	1812	13
8:45	1692	16	1860	14	1956	12
8:50	1776	20	2028	17	1980	14
8:55	1764	17	1608	11	1512	9
9:00	1992	20	1656	15	1836	13
9:05	1656	17	1860	16	2004	14
9:10	1452	15	1464	10	1260	7
9:15	1476	13	1212	8	1284	6
9:20	1320	11	1392	8	1080	6
9:25	1428	12	1488	10	1116	6
9:30	1488	13	1440	8	1224	6
9:35	1212	11	1548	9	1152	6
9:40	1404	13	1500	10	1236	6
9:45	1380	12	1404	9	1476	8
9:50	1248	10	1440	9	1056	5
9:55	1380	11	1536	9	1128	5
10:00	1392	11	1320	10	1056	5

DETECTOR STATION 185

detector 854		detector 855		detector 856	
VOL	OCC	VOL	OCC	VOL	OCC
2028	16	2232	15	2364	14
1956	15	2124	14	2544	15
1824	13	1992	12	2172	11
1860	14	1800	11	2376	13
2124	16	2052	14	2292	13
1968	16	2232	15	2340	14
1860	14	2040	14	2340	14
1824	17	2016	19	2304	18
1968	33	1752	32	1896	27
1968	24	1980	22	2052	19
1872	31	1704	30	1920	28
1848	28	1836	24	2040	24
1836	26	2016	25	2112	20
2052	26	1956	20	2232	18
1560	22	1728	19	1992	17
1944	28	1860	20	2160	17
2064	20	1980	18	2220	16
1872	32	1932	26	2292	21
1848	31	1860	25	2292	23
1908	24	1920	26	2220	20
1716	13	1968	13	2112	12
1872	14	2040	13	2232	12
1764	13	1800	12	1920	11
1788	13	1644	10	1836	11
1692	13	1692	11	2040	12
1560	12	1716	10	1656	9
1212	8	1356	7	1332	6
1164	9	1500	9	1152	6
1272	9	1596	10	1320	7
1284	9	1452	8	1260	6
1404	10	1644	9	1152	6
1104	8	1632	10	1308	6
1272	8	1740	10	1548	7
1164	8	1440	8	1284	6
1536	10	1512	8	1152	5
1284	8	1524	9	1092	5
1260	8	1536	9	1224	6

DETECTOR STATION 186

TIME	detector 843		detector 844		detector 845	
	VOL	OCC	VOL	OCC	VOL	OCC
7:00	2220	16	2040	15	2328	17
7:05	2208	17	2184	15	2328	17
7:10	2196	16	1920	14	2148	14
7:15	1980	15	1944	13	2316	15
7:20	2304	18	1932	15	2412	17
7:25	2388	19	1992	15	2400	18
7:30	2220	21	1980	18	2232	19
7:35	2112	23	1932	20	1980	26
7:40	2208	21	1956	17	2064	18
7:45	2076	19	1884	17	2136	20
7:50	2112	24	1740	21	1764	29
7:55	2244	19	1848	15	2016	16
8:00	2016	25	1980	23	2064	25
8:05	2328	26	1920	20	2196	23
8:10	1920	26	1512	25	1968	27
8:15	2232	19	1872	15	2148	16
8:20	2268	20	1980	17	2148	17
8:25	2076	18	1920	16	2196	16
8:30	2148	20	1860	15	2196	16
8:35	2256	19	1920	16	2376	17
8:40	2172	17	1764	14	2112	14
8:45	2232	16	1836	14	2160	14
8:50	2148	16	1740	12	2040	13
8:55	1884	14	1728	12	1680	11
9:00	1908	15	1656	13	1872	13
9:05	2004	15	1572	11	1704	11
9:10	1368	10	1344	9	1296	8
9:15	1332	10	1320	9	1092	6
9:20	1512	10	1392	11	1428	9
9:25	1560	11	1428	9	1212	7
9:30	1560	11	1536	10	1296	8
9:35	1524	10	1308	9	1272	7
9:40	1596	11	1596	11	1368	9
9:45	1368	10	1344	9	1236	7
9:50	1812	13	1308	8	1320	7
9:55	1632	12	1428	10	1116	6
10:00	1560	10	1344	9	1152	7

DETECTOR STATION 187

detector 826		detector 827		detector 828	
VOL	OCC	VOL	OCC	VOL	OCC
1164	6	1860	12	2280	14
1380	8	1860	14	2340	21
1224	8	1896	16	2256	17
1344	7	1668	10	2148	12
1404	8	1836	13	2256	15
1392	8	1944	16	2268	20
1548	10	1764	18	1932	18
1404	11	1764	18	1956	22
1380	11	1824	22	2016	20
1200	8	1776	24	1848	23
1380	10	1668	25	1860	22
1608	17	1572	17	1848	27
1476	14	1668	18	1884	27
1476	14	1716	20	1824	28
1620	12	1608	16	1980	14
1272	7	1812	11	2088	11
1296	7	1800	12	2112	13
1416	7	1752	11	1920	10
1392	7	1680	11	2016	11
1428	8	1620	9	2088	11
1140	6	1668	10	2028	11
1020	5	1848	11	2076	10
1020	5	1788	10	1896	10
804	4	1692	9	1512	8
924	5	1680	10	1716	9
852	4	1728	10	1572	8
804	4	1428	7	1152	6
708	3	1212	7	1116	5
564	2	1428	8	1416	8
864	4	1308	7	1080	5
660	3	1416	8	1332	7
600	3	1404	8	1260	6
708	3	1500	9	1332	7
792	4	1380	8	1104	5
696	3	1344	7	1092	5
588	3	1356	8	1032	4
636	3	1356	7	1044	6

DETECTOR STATION 188

TIME	detector 831		detector 832		detector 833	
	VOL	OCC	VOL	OCC	VOL	OCC
7:00	1368	9	2208	22	1836	17
7:05	1284	8	2352	18	2100	18
7:10	1548	11	2304	17	1956	16
7:15	1596	10	2316	17	1776	15
7:20	1632	13	2088	20	1872	18
7:25	1608	11	2148	24	2052	22
7:30	1668	12	2148	19	2016	21
7:35	1560	11	2100	19	2064	22
7:40	1428	11	2124	22	1980	23
7:45	1524	12	1992	24	1944	26
7:50	1440	10	2172	21	1968	22
7:55	1644	14	2040	25	1728	24
8:00	1488	12	2016	22	1848	27
8:05	1752	16	2016	24	1872	27
8:10	1836	16	2172	21	1908	24
8:15	1656	12	2208	17	1944	18
8:20	1416	10	2124	19	2016	23
8:25	1764	14	2064	19	1800	23
8:30	1632	10	2160	16	1860	17
8:35	1620	9	2076	11	1884	12
8:40	1452	8	2076	11	1920	12
8:45	1404	8	1968	10	1908	12
8:50	1368	8	2088	11	1800	11
8:55	1212	6	1524	8	1740	10
9:00	1188	7	1776	9	1800	12
9:05	1332	8	1728	9	1680	11
9:10	1140	6	1380	7	1632	10
9:15	1020	6	1236	6	1416	8
9:20	924	5	1404	7	1488	9
9:25	1140	6	1224	6	1332	7
9:30	1056	6	1380	7	1572	9
9:35	888	4	1332	6	1536	9
9:40	1032	5	1368	7	1452	9
9:45	1020	5	1272	6	1464	9
9:50	1092	6	1104	5	1368	8
9:55	1020	5	1080	5	1416	8
10:00	1044	5	948	4	1452	8

DETECTOR STATION 189

detector 815		detector 816	
VOL	OCC	VOL	OCC
1872	12	2256	13
2136	15	2376	14
2112	15	2352	14
1860	13	2352	15
1836	13	2316	14
1968	14	2304	14
1992	14	2328	14
2088	17	2256	15
2028	18	2316	17
2124	20	2184	18
1944	16	2376	18
1824	20	2304	21
1992	19	2100	16
2040	18	2268	17
2148	18	2280	16
2076	20	2280	17
2040	17	2208	14
1884	15	2268	14
2004	15	2292	14
1848	12	2076	11
1896	11	1968	10
1992	13	2184	12
1812	11	2064	11
1656	9	1668	8
1836	12	1836	10
1752	11	1860	10
1500	9	1452	8
1416	8	1308	6
1536	9	1380	7
1464	8	1200	6
1548	9	1680	8
1572	9	1404	7
1476	8	1332	7
1296	7	1368	6
1464	8	1212	5
1344	7	1200	5
1368	7	1116	5

DETECTOR STATION 190

TIME	detector 808		detector 809		detector 810	
	VOL	OCC	VOL	OCC	VOL	OCC
7:00	456	2	1488	9	2352	12
7:05	504	2	1596	10	2460	13
7:10	468	2	1608	10	2376	12
7:15	468	2	1608	10	2376	12
7:20	576	3	1416	9	2328	12
7:25	504	3	1836	11	2388	13
7:30	504	3	1836	11	2388	13
7:35	456	2	1728	11	2376	13
7:40	492	2	1764	12	2424	14
7:45	492	2	1764	12	2424	14
7:50	396	2	1776	11	2472	14
7:55	504	3	1668	11	2364	14
8:00	672	4	1560	10	2316	13
8:05	756	4	1584	11	2220	12
8:10	756	4	1584	11	2220	12
8:15	564	3	1500	10	2412	14
8:20	576	3	1572	10	2196	11
8:25	576	3	1572	10	2196	11
8:30	732	4	1368	9	2040	11
8:35	384	1	1428	9	2028	10
8:40	480	2	1308	8	1980	10
8:45	480	2	1308	8	1980	10
8:50	504	2	1500	9	2268	12
8:55	444	2	1464	9	2016	10
9:00	360	1	1452	8	1776	8
9:05	276	1	1380	8	1800	9
9:10	420	2	1272	8	2004	11
9:15	288	1	1356	7	1620	8
9:20	384	2	1140	6	1428	7
9:25	252	1	1212	8	1500	7
9:30	408	2	1008	5	1284	6
9:35	252	1	1272	8	1668	8
9:40	216	1	1308	8	1572	7
9:45	264	1	1332	8	1440	7
9:50	240	1	1128	7	1548	7
9:55	288	1	1068	6	1380	6
10:00	336	1	1248	7	1380	6

DETECTOR STATION 191

detector 801		detector 802	
VOL	OCC	VOL	OCC
1632	11	2256	12
1632	10	2328	12
1596	11	2352	12
1680	12	2292	12
1632	11	2304	12
1524	10	2148	10
1980	14	2232	12
1728	11	2232	11
1800	13	2280	12
1884	12	2280	11
1908	14	2292	13
1920	12	2328	12
1812	12	2256	12
1740	11	2268	11
1848	13	2112	11
1656	11	1932	9
1596	11	2316	12
1680	12	2100	10
1440	10	1836	9
1584	12	2172	10
1500	11	1944	9
1440	10	1884	9
1488	12	2028	9
1656	12	1824	9
1524	11	1596	7
1392	10	1620	8
1488	11	1716	9
1464	10	1584	8
1200	8	1344	6
1188	9	1224	6
1020	7	1308	7
1284	9	1524	7
1488	10	1392	7
1200	9	1440	7
1296	9	1476	7
1224	8	1284	6
1332	9	1152	5

DETECTOR STATION 480

DETECTOR STATION 481

TIME	detector 1209		detector 1210	
	VOL	OCC	VOL	OCC
7:00	1704	13	2208	16
7:05	1800	15	2040	15
7:10	1788	14	2268	16
7:15	1800	15	2160	15
7:20	1608	13	2040	14
7:25	2028	17	2088	15
7:30	1896	15	2220	15
7:35	1836	17	2052	15
7:40	2052	16	2136	16
7:45	2052	18	2208	16
7:50	1944	15	2280	16
7:55	1932	15	2124	15
8:00	1860	15	2112	15
8:05	1896	16	2112	15
8:10	1728	14	1848	12
8:15	1788	15	2004	14
8:20	1728	15	2076	14
8:25	1596	14	1680	11
8:30	1800	16	2016	14
8:35	1644	13	1836	12
8:40	1656	14	1632	11
8:45	1692	14	1836	12
8:50	1704	15	1860	13
8:55	1572	13	1512	10
9:00	1596	14	1404	10
9:05	1548	13	1608	11
9:10	1548	12	1356	10
9:15	1332	11	1356	9
9:20	1284	11	1020	7
9:25	1176	10	1224	9
9:30	1452	12	1260	9
9:35	1464	12	1320	9
9:40	1320	11	1284	10
9:45	1380	12	1392	10
9:50	1428	11	1056	7
9:55	1392	11	1080	7
10:00	1320	11	996	7

detector 1211		detector 1212	
VOL	OCC	VOL	OCC
2148	18	1812	13
1896	15	1776	12
1992	15	2160	15
2016	18	1824	12
1884	16	1836	12
2184	18	1896	12
2112	18	2064	14
2040	19	1860	13
2232	18	1944	13
2208	18	2076	14
2196	17	1956	13
2220	19	1908	13
2184	19	1848	12
2148	19	1776	12
1968	15	1680	11
2064	17	1680	11
2076	18	1800	11
1956	16	1392	9
1920	17	1680	11
2016	16	1656	11
1788	15	1392	9
1884	15	1692	11
1992	17	1560	10
1800	15	1344	9
1860	15	1200	8
1656	14	1368	9
1836	14	1188	8
1560	12	1104	7
1548	13	900	6
1476	13	900	6
1656	12	1140	7
1572	13	1164	8
1500	12	1188	8
1548	13	1272	9
1476	12	900	6
1596	12	804	5
1512	13	924	6

DETECTOR STATION 482

DETECTOR STATION 483

TIME	detector 1214		detector 1215		detector 1217		detector 1218	
	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
7:00	1224	4	1320	4	1092	8	948	6
7:05	1116	4	1392	5	924	6	900	6
7:10	1428	5	1500	5	1044	7	1032	6
7:15	1272	4	1416	5	1056	8	1080	7
7:20	1092	4	1332	5	1044	8	1056	6
7:25	1284	4	1248	4	876	6	960	6
7:30	1152	4	1524	5	1104	7	1032	6
7:35	1200	4	1380	5	936	7	1080	6
7:40	1236	4	1368	5	1020	7	1008	6
7:45	1236	4	1524	5	1008	7	1152	7
7:50	1344	5	1332	5	972	6	1104	7
7:55	1116	4	1368	5	912	6	1080	6
8:00	1284	4	1260	4	816	5	1008	6
8:05	1152	4	1092	4	876	6	900	5
8:10	1116	4	1176	4	888	6	864	5
8:15	1128	4	1212	4	828	6	960	6
8:20	1344	5	1188	4	996	7	816	5
8:25	1032	3	1116	4	996	8	912	5
8:30	1032	3	1056	4	828	6	720	4
8:35	1188	4	1152	4	972	7	864	5
8:40	1080	4	1044	3	960	7	804	5
8:45	1116	4	1272	4	984	8	1020	7
8:50	1092	4	1188	4	996	7	960	6
8:55	1092	4	888	3	888	6	600	4
9:00	1032	3	744	2	744	5	576	3
9:05	1092	4	924	3	804	6	504	3
9:10	1080	4	936	3	852	6	660	5
9:15	888	3	840	3	780	5	456	3
9:20	888	3	612	2	816	7	468	3
9:25	888	3	708	2	828	7	588	4
9:30	924	3	840	3	732	5	480	3
9:35	948	3	948	3	828	6	648	4
9:40	948	3	840	3	768	5	708	5
9:45	1044	3	1032	3	960	8	732	5
9:50	948	3	744	2	780	6	528	3
9:55	816	3	516	1	732	5	348	2
10:00	900	3	624	2	648	5	408	2

DETECTOR STATION 484

DETECTOR STATION 485

TIME	detector 1224		detector 1225	
	VOL	OCC	VOL	OCC
7:00	648	6	828	6
7:05	600	5	732	5
7:10	576	5	852	6
7:15	708	6	1116	8
7:20	708	7	972	7
7:25	588	5	984	7
7:30	540	4	912	6
7:35	792	7	1128	8
7:40	696	7	1056	8
7:45	828	7	1140	8
7:50	804	7	1176	9
7:55	732	6	1044	7
8:00	696	6	864	6
8:05	576	5	912	6
8:10	600	5	876	6
8:15	516	5	852	6
8:20	636	5	840	6
8:25	708	7	960	6
8:30	576	6	732	5
8:35	600	5	816	6
8:40	624	7	840	6
8:45	672	7	768	6
8:50	720	6	1032	7
8:55	564	5	564	4
9:00	612	7	588	4
9:05	516	5	348	2
9:10	624	5	576	5
9:15	564	5	468	3
9:20	516	4	492	4
9:25	684	7	528	4
9:30	504	5	432	3
9:35	648	6	708	5
9:40	624	5	540	5
9:45	660	7	672	5
9:50	624	7	540	4
9:55	576	5	348	2
10:00	480	4	264	1

detector 1228		detector 1229		detector 1230	
VOL	OCC	VOL	OCC	VOL	OCC
12	0	624	5	768	5
36	0	588	5	672	4
12	0	768	6	696	5
24	0	828	7	984	7
60	0	624	6	936	6
24	0	792	6	912	6
0	0	588	4	828	5
48	0	888	7	1008	7
36	0	780	7	1020	7
36	0	912	7	1020	7
12	0	864	7	1056	7
36	0	708	6	996	7
0	0	720	6	960	6
12	0	732	6	756	5
48	0	660	5	768	5
72	0	696	6	756	5
108	1	720	6	864	6
72	0	768	7	852	5
48	0	660	6	648	4
72	0	624	5	804	6
84	0	720	7	744	5
72	0	804	8	624	5
24	0	828	7	876	6
48	0	672	6	492	4
72	0	684	6	480	3
72	0	540	5	348	2
24	0	672	6	480	4
24	0	528	5	456	3
36	0	528	4	456	3
36	0	672	6	420	3
60	0	564	5	492	4
48	0	684	6	624	4
24	0	636	5	540	4
84	1	660	7	600	5
60	0	648	6	492	3
96	1	636	6	324	2
48	0	540	5	228	1

DETECTOR STATION 486

DETECTOR STATION 487

TIME	detector 1231		detector 1232		detector 1234		detector 1235	
	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
7:00	936	7	816	5	564	5	924	7
7:05	960	7	624	4	708	6	804	6
7:10	1032	7	936	5	588	5	780	6
7:15	1104	8	1020	6	672	5	828	6
7:20	1080	8	1152	7	720	8	1236	9
7:25	1164	8	1092	6	720	8	1236	9
7:30	1152	8	1008	6	804	7	1056	8
7:35	1404	10	1080	6	804	7	1056	8
7:40	1380	10	1128	7	924	8	1224	9
7:45	1380	9	1392	8	888	7	1524	11
7:50	1356	9	1272	7	924	8	1812	14
7:55	1320	9	1260	7	804	7	1320	10
8:00	1200	8	1164	7	636	6	1176	9
8:05	1308	9	1164	7	756	7	1320	10
8:10	1092	7	900	5	660	6	948	7
8:15	1176	8	852	5	660	6	948	7
8:20	1164	9	984	6	804	7	948	7
8:25	1140	8	816	4	804	7	1044	8
8:30	936	7	792	4	756	7	840	6
8:35	1044	8	852	5	720	7	756	5
8:40	1104	10	792	4	720	7	756	5
8:45	1092	9	708	5	780	8	828	6
8:50	1128	8	924	5	924	8	924	7
8:55	912	7	612	4	924	8	924	7
9:00	912	7	540	3	660	7	492	3
9:05	756	6	396	2	672	7	516	4
9:10	996	8	624	4	624	7	552	4
9:15	792	6	456	2	648	7	624	5
9:20	852	6	540	3	588	5	504	4
9:25	840	7	468	3	648	6	480	4
9:30	792	7	552	3	648	6	480	4
9:35	888	7	636	3	612	7	672	5
9:40	792	6	576	3	708	6	516	4
9:45	936	8	612	4	744	7	684	5
9:50	804	6	528	3	588	6	468	4
9:55	948	7	336	2	672	7	708	5
10:00	648	5	276	1	672	6	516	3



DETECTOR STATION 488

DETECTOR STATION 511

TIME	detector 1237		detector 1238		detector 1840		detector 1841	
	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
7:00	924	7	996	6	936	5	960	4
7:05	900	6	840	5	1020	5	804	3
7:10	1020	7	984	6	1020	5	996	4
7:15	972	7	1236	7	1080	5	1104	4
7:20	1212	9	1464	9	1260	6	1344	5
7:25	996	6	1380	8	1032	5	1356	5
7:30	1020	7	1464	8	1080	5	1380	6
7:35	1296	9	1800	11	1284	6	1608	6
7:40	1140	9	1476	9	1224	7	1404	6
7:45	1308	9	1860	11	1308	7	1716	7
7:50	1320	9	2352	14	1428	8	2124	9
7:55	1320	9	1836	11	1344	7	1740	7
8:00	1092	7	1512	9	1188	6	1536	6
8:05	1140	8	1584	9	1164	6	1368	5
8:10	1008	7	1368	8	1152	6	1212	5
8:15	1152	8	1224	7	1212	6	1368	5
8:20	1236	9	1668	11	1368	7	1368	6
8:25	1104	8	1524	9	1140	6	1476	6
8:30	1032	8	1188	7	1104	6	1068	4
8:35	936	7	1044	7	1008	5	972	4
8:40	1056	9	1140	6	1248	7	960	4
8:45	1068	9	1188	8	1212	6	1224	6
8:50	1212	9	1404	9	1212	7	1272	6
8:55	1056	8	1080	7	1116	6	1152	5
9:00	960	7	780	5	948	5	876	4
9:05	924	7	780	4	900	5	780	3
9:10	888	7	612	4	960	5	588	2
9:15	960	8	768	5	1056	7	780	3
9:20	804	6	564	3	888	5	504	2
9:25	876	7	684	5	852	5	636	2
9:30	828	7	636	4	924	5	624	2
9:35	828	6	768	4	1032	5	576	2
9:40	1152	8	972	6	1044	6	900	3
9:45	768	6	660	4	780	4	516	2
9:50	1020	8	1020	6	1212	6	996	4
9:55	876	7	756	4	984	5	672	2
10:00	924	6	552	3	936	4	396	1

DETECTOR STATION 512

DETECTOR STATION 513

TIME	detector 1843		detector 1844		detector 1848		detector 1849	
	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
7:00	636	5	1116	8	1272	9	1320	9
7:05	528	4	1200	8	1332	9	1380	10
7:10	444	4	1248	8	1452	10	1296	9
7:15	672	5	1416	9	1380	9	1476	10
7:20	744	7	1752	12	1692	13	1944	14
7:25	684	6	1764	12	1560	12	1812	12
7:30	804	7	1548	10	1596	12	1716	12
7:35	816	7	1824	12	1584	11	1848	12
7:40	864	7	1668	11	1632	12	1908	13
7:45	1008	9	2052	15	1752	13	2172	15
7:50	1056	9	2256	16	1860	13	2484	17
7:55	984	8	1932	14	1788	13	2172	15
8:00	840	8	1944	13	1632	12	2112	14
8:05	876	8	1656	12	1416	10	1632	11
8:10	780	6	1416	10	1572	11	1692	12
8:15	864	7	1668	12	1452	10	1620	11
8:20	948	8	1404	10	1404	10	1560	11
8:25	960	7	1524	11	1440	11	1824	13
8:30	840	9	1380	10	1344	10	1464	10
8:35	768	7	1332	11	1128	9	1368	10
8:40	924	8	1152	8	1416	10	1344	10
8:45	936	9	1404	10	1368	11	1500	11
8:50	816	9	1392	11	1224	9	1404	11
8:55	888	7	1296	9	1488	11	1692	12
9:00	888	7	876	7	1092	8	1020	8
9:05	720	6	900	7	1008	8	1044	7
9:10	648	6	708	5	984	9	708	5
9:15	804	8	996	8	1188	10	972	7
9:20	708	6	564	3	1164	8	768	6
9:25	720	6	768	6	1080	8	900	6
9:30	636	6	852	6	1188	10	900	6
9:35	792	6	708	5	1104	9	816	6
9:40	684	7	1128	8	1068	8	960	6
9:45	648	5	768	5	1044	8	900	6
9:50	1056	10	864	6	1164	9	900	7
9:55	792	8	840	6	1008	7	828	6
10:00	708	5	540	4	1152	9	708	5

DETECTOR STATION 514

DETECTOR STATION 515

TIME	detector 1851		detector 1852		detector 1854		detector 1855	
	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
7:00	1068	8	1440	9	1380	8	1404	7
7:05	984	7	1428	9	1476	9	1368	7
7:10	1104	8	1320	8	1452	9	1392	8
7:15	1044	7	1752	11	1464	9	1740	9
7:20	1140	9	1836	12	1764	11	1980	11
7:25	1176	10	1980	13	1692	12	2100	12
7:30	1272	10	1860	12	1692	11	1968	10
7:35	1164	8	1908	12	1632	10	1896	10
7:40	1080	9	1884	12	1728	11	2004	11
7:45	1356	11	2208	15	1584	11	2172	12
7:50	1476	11	2376	16	1836	12	2688	16
7:55	1296	10	2424	16	1860	17	2280	18
8:00	1260	14	1932	19	1596	22	1920	29
8:05	1164	14	1488	22	1596	18	1824	24
8:10	1236	11	1728	15	1572	20	1800	21
8:15	1104	8	1668	11	1776	14	2160	14
8:20	1248	10	1596	11	1572	10	1740	10
8:25	1236	10	1824	12	1572	10	1944	11
8:30	1176	9	1476	9	1512	10	1524	7
8:35	996	8	1344	9	1392	9	1380	8
8:40	1152	9	1332	9	1524	10	1476	8
8:45	1188	10	1368	9	1464	10	1452	8
8:50	1020	8	1416	10	1416	9	1536	9
8:55	1224	11	1764	11	1536	10	1716	9
9:00	960	7	1056	8	1224	8	1092	6
9:05	876	8	1032	7	1128	7	996	6
9:10	840	8	696	4	1152	8	828	5
9:15	1188	10	972	7	1272	9	1116	6
9:20	936	7	720	5	1200	8	840	5
9:25	984	8	828	6	1260	8	900	5
9:30	924	8	972	7	1344	9	1020	6
9:35	1008	8	876	6	1296	9	924	5
9:40	1032	8	900	5	1164	7	912	5
9:45	948	8	888	6	1188	8	1044	6
9:50	912	8	888	6	1140	7	912	5
9:55	840	6	804	5	1128	7	960	5
10:00	1056	8	756	5	1176	7	864	5

DETECTOR STATION 516

DETECTOR STATION 517

TIME	detector 1856		detector 1857		detector 1858		detector 1859	
	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
7:00	1704	12	1104	7	948	7	996	7
7:05	1692	12	1032	7	960	8	1116	8
7:10	1848	13	1164	8	792	6	876	6
7:15	1812	12	1392	9	816	6	936	7
7:20	1920	14	1656	12	960	7	1116	8
7:25	2040	17	1824	13	1044	8	1416	10
7:30	1896	14	1668	12	1068	10	1656	13
7:35	1848	19	1704	14	984	7	1404	9
7:40	1920	18	1908	16	984	9	1560	12
7:45	1956	15	1716	11	1260	11	1716	13
7:50	2100	15	2244	16	1296	12	1572	11
7:55	1788	19	2052	18	1332	10	2088	15
8:00	1716	18	1668	18	1320	11	1728	12
8:05	1680	20	1836	19	1104	9	1356	11
8:10	1968	24	1872	20	1176	11	1608	12
8:15	2196	17	1848	13	1248	11	1536	12
8:20	1932	13	1404	9	1116	10	1668	12
8:25	1860	14	1608	11	1116	9	1116	8
8:30	1740	12	1440	9	1224	10	1452	11
8:35	1548	11	1176	7	1140	9	1284	9
8:40	1740	13	1296	9	936	7	888	6
8:45	1668	13	1104	7	936	8	1188	9
8:50	1692	12	1284	9	1008	8	840	6
8:55	1812	14	1368	9	972	9	1152	9
9:00	1500	11	912	6	1164	10	1200	8
9:05	1332	10	816	6	936	8	816	6
9:10	1296	10	648	4	1032	8	696	5
9:15	1344	11	756	5	756	7	612	4
9:20	1488	10	840	6	888	9	780	5
9:25	1416	10	660	4	924	7	756	7
9:30	1584	12	888	7	696	5	684	5
9:35	1404	11	696	5	792	7	708	5
9:40	1356	10	720	5	852	8	648	5
9:45	1404	11	876	6	876	7	696	5
9:50	1248	10	816	6	936	8	804	6
9:55	1260	9	780	5	660	6	744	6
10:00	1248	9	696	4	828	7	624	5

DETECTOR STATION 518

TIME	detector 1861		detector 1862	
	VOL	OCC	VOL	OCC
7:00	876	7	960	6
7:05	816	6	804	5
7:10	876	6	756	5
7:15	888	5	924	6
7:20	1020	7	1164	7
7:25	1080	8	1440	9
7:30	1032	6	1200	7
7:35	1176	8	1284	7
7:40	1152	8	1512	9
7:45	1200	9	1272	8
7:50	1464	10	1752	11
7:55	1140	8	1464	9
8:00	1116	8	1284	8
8:05	1212	9	1332	8
8:10	1152	9	1320	8
8:15	888	6	1464	9
8:20	888	6	984	6
8:25	1176	8	1176	8
8:30	900	6	1140	7
8:35	876	6	792	5
8:40	804	6	1044	7
8:45	876	7	924	6
8:50	840	7	936	7
8:55	840	7	936	7
9:00	864	6	720	5
9:05	972	7	612	4
9:10	516	4	624	4
9:15	720	6	624	4
9:20	804	6	672	5
9:25	600	4	564	3
9:30	720	5	648	5
9:35	708	6	480	3
9:40	792	6	660	4
9:45	792	6	660	4
9:50	864	7	732	4
9:55	552	4	624	4
10:00	696	5	648	5

	detector 1595		detector 1596		detector 1600		detector 1601		detector 1602		detector 1608	
TIME	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
7:00	396	3	528	3	1032	6	468	2	792	6	504	4
7:05	396	3	576	3	852	5	540	3	180	1	588	5
7:10	312	3	540	3	948	5	624	3	300	3	432	3
7:15	456	3	552	3	1080	6	480	2	336	4	312	2
7:20	360	2	624	4	1224	7	408	2	168	2	456	3
7:25	384	3	804	5	1452	9	600	3	192	3	744	6
7:30	264	2	972	6	1356	7	384	2	156	2	576	4
7:35	240	1	792	5	1188	6	444	2	156	3	888	7
7:40	228	1	648	4	1140	6	432	2	180	4	528	4
7:45	300	2	600	3	1284	7	468	2	156	3	660	6
7:50	228	2	840	5	1332	8	552	3	168	3	864	7
7:55	312	2	1020	6	1512	9	576	3	132	2	1104	9
8:00	252	1	756	4	1368	7	480	2	144	2	1212	11
8:05	168	1	996	6	1152	6	588	3	180	3	1104	9
8:10	228	1	888	5	1032	6	588	3	132	2	888	7
8:15	312	2	852	5	948	5	552	3	108	1	660	5
8:20	360	2	708	4	840	4	432	2	156	3	780	6
8:25	204	1	768	5	960	5	492	2	132	2	816	6
8:30	216	1	780	5	1104	6	636	3	168	2	564	4
8:35	168	1	672	4	816	4	660	3	192	3	648	5
8:40	264	2	552	3	888	5	648	4	204	4	672	6
8:45	120	1	372	2	864	5	684	4	384	7	696	6
8:50	156	1	312	2	876	5	672	4	324	4	624	5
8:55	204	1	372	2	1224	7	612	3	168	1	576	5
9:00	360	3	360	2	900	5	708	4	132	1	720	6
9:05	204	2	264	1	588	3	720	4	288	2	564	4
9:10	120	1	252	1	684	3	684	4	336	3	552	5
9:15	144	1	216	1	540	3	516	3	252	2	516	4
9:20	180	1	300	1	588	3	444	2	192	1	324	3
9:25	108	1	144	0	612	3	492	3	240	2	552	4
9:30	144	1	180	1	672	3	480	2	204	2	552	5
9:35	120	1	168	1	636	3	300	1	360	3	516	4
9:40	204	1	168	1	612	3	588	3	264	2	516	4
9:45	276	2	204	1	540	3	516	3	312	3	504	4
9:50	72	0	192	1	516	2	552	3	180	1	540	4
9:55	156	1	168	1	516	2	396	2	264	2	456	3
10:00	168	1	204	1	552	3	384	2	216	2	720	6

	detector 1613		detector 1612		detector 1614		detector 1618		detector 1487		detector 1619	
TIME	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
7:00	336	3	144	1	12	0	108	1	480	7	0	100
7:05	204	1	180	2	12	0	156	2	264	3	0	100
7:10	264	2	60	0	0	0	132	1	648	9	0	100
7:15	312	2	144	3	24	0	132	1	264	3	0	100
7:20	300	2	228	8	24	1	192	2	456	6	0	100
7:25	228	2	180	7	12	0	216	2	408	5	0	100
7:30	228	1	108	2	0	0	60	0	192	2	0	99
7:35	180	1	168	3	36	0	144	1	240	3	0	100
7:40	156	1	132	3	0	0	144	1	228	3	0	100
7:45	228	2	144	4	24	0	156	2	276	3	0	100
7:50	192	1	168	4	72	2	144	2	444	6	0	100
7:55	288	3	168	2	0	0	108	1	372	4	0	99
8:00	192	1	156	4	24	0	120	1	528	7	0	100
8:05	96	1	144	4	12	0	132	1	228	2	0	100
8:10	168	1	156	5	36	1	120	1	204	2	0	100
8:15	156	1	144	4	60	2	108	1	240	3	0	100
8:20	84	0	180	4	60	1	72	0	228	3	0	100
8:25	72	0	168	4	24	0	96	1	312	4	0	100
8:30	264	2	180	5	12	0	84	1	228	3	0	100
8:35	252	2	144	4	48	4	72	1	216	3	0	99
8:40	156	1	156	4	12	0	0	0	288	4	0	100
8:45	228	2	468	13	12	0	36	0	192	3	0	100
8:50	288	2	396	8	60	1	48	0	288	5	0	100
8:55	348	3	264	2	12	0	48	0	312	4	0	100
9:00	228	2	252	1	0	0	36	0	276	3	0	100
9:05	252	2	312	3	0	0	48	0	264	4	0	100
9:10	264	3	228	1	0	0	48	0	264	4	0	100
9:15	192	1	324	4	24	0	0	0	204	2	0	100
9:20	228	2	300	2	12	0	48	0	168	2	0	100
9:25	192	1	120	0	24	0	12	0	252	4	0	100
9:30	348	3	240	2	12	0	12	0	168	2	0	100
9:35	204	1	312	3	24	0	36	0	108	1	0	100
9:40	264	2	360	4	0	0	24	0	252	3	0	99
9:45	300	2	264	2	24	0	36	0	204	2	0	100
9:50	192	1	324	3	24	0	12	0	240	4	0	100
9:55	168	1	480	5	36	0	60	0	228	3	0	100
10:00	228	2	216	1	12	0	24	0	252	3	0	100

	detector 1620		detector 998		detector 894		detector 882		detector 881		detector 872	
TIME	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
7:00	132	1	300	2	0	100	360	2	264	1	240	2
7:05	84	1	264	2	0	99	408	2	168	1	216	1
7:10	144	3	348	3	0	100	312	1	156	1	252	1
7:15	156	3	252	1	0	100	252	1	192	2	324	2
7:20	108	4	276	2	0	100	312	1	180	2	300	2
7:25	108	2	396	3	0	100	360	2	156	1	312	2
7:30	96	2	552	4	0	100	204	1	108	1	384	2
7:35	144	3	816	6	0	100	240	1	72	1	336	2
7:40	96	2	540	4	0	100	108	0	108	1	300	2
7:45	120	2	612	5	0	100	216	1	36	0	300	2
7:50	96	2	924	9	0	100	180	1	108	1	336	2
7:55	72	1	852	7	0	100	252	1	60	0	264	1
8:00	108	2	756	6	0	100	252	1	120	1	300	2
8:05	96	2	684	5	0	100	264	1	84	1	264	1
8:10	120	2	396	3	0	100	324	2	96	1	360	3
8:15	96	2	312	2	0	100	312	1	72	1	348	2
8:20	120	3	264	2	0	100	348	2	84	1	312	2
8:25	108	2	252	1	0	100	144	0	96	0	324	2
8:30	108	3	300	2	0	100	156	0	48	0	348	2
8:35	120	5	252	2	0	99	228	1	96	1	312	2
8:40	252	5	324	2	0	100	132	0	72	0	264	1
8:45	240	5	288	2	0	100	264	1	240	2	264	1
8:50	264	2	312	2	0	100	264	1	288	2	228	1
8:55	240	2	372	3	0	100	444	3	132	1	432	3
9:00	180	1	264	2	0	100	372	2	240	1	660	6
9:05	168	1	336	3	0	100	288	1	324	2	408	2
9:10	180	1	324	2	0	100	180	1	156	1	300	2
9:15	228	2	336	3	0	100	192	1	312	2	252	2
9:20	144	1	216	1	0	99	240	1	384	2	264	1
9:25	228	2	312	2	0	100	216	1	360	2	192	1
9:30	192	1	324	2	0	100	180	1	300	2	156	1
9:35	228	2	240	1	0	100	252	1	444	3	216	1
9:40	336	3	312	2	0	100	108	0	324	2	204	1
9:45	216	2	228	1	0	99	252	1	300	2	276	1
9:50	144	1	456	4	0	100	204	1	348	2	324	2
9:55	204	1	276	2	0	100	204	1	240	1	300	2
10:00	240	2	360	2	0	100	192	1	312	2	312	2



	detector 871		detector 121		detector 162		detector 104		detector 867		detector 858	
TIME	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
7:00	288	3	276	4	1464	27	528	7	384	4	384	2
7:05	144	1	156	2	1344	28	420	6	600	8	576	4
7:10	180	3	288	3	1212	21	324	5	492	4	492	3
7:15	252	6	180	3	1380	28	348	5	576	6	468	3
7:20	252	7	204	4	1260	20	384	4	372	3	576	4
7:25	180	4	216	3	1344	37	360	5	660	9	480	8
7:30	180	4	96	1	1368	47	420	7	600	8	444	3
7:35	156	4	216	3	1320	47	456	5	432	4	600	4
7:40	168	4	204	2	1140	56	348	4	552	8	468	3
7:45	168	4	216	2	1032	52	204	2	576	10	408	2
7:50	144	3	216	3	1116	47	216	2	624	11	516	3
7:55	156	3	120	2	816	54	312	4	540	10	516	4
8:00	144	3	324	6	1044	54	264	3	540	10	420	3
8:05	168	5	192	2	996	53	228	3	660	12	468	3
8:10	132	4	204	3	1032	52	336	6	576	11	528	3
8:15	156	5	336	7	948	54	276	4	612	12	600	4
8:20	156	4	360	6	1116	52	288	3	648	12	636	5
8:25	156	5	372	8	1188	49	336	4	1032	16	732	6
8:30	156	3	372	8	1188	49	396	6	708	6	648	4
8:35	156	6	300	5	1200	49	444	5	900	9	696	4
8:40	156	4	360	6	1488	35	456	5	876	7	468	3
8:45	468	13	360	6	1488	35	372	6	480	3	516	3
8:50	360	6	396	8	1296	50	624	10	864	7	528	3
8:55	240	3	408	7	1380	30	420	6	660	5	492	5
9:00	276	3	420	7	1236	25	372	5	576	5	648	4
9:05	360	5	204	2	1188	38	612	12	576	5	456	3
9:10	288	3	204	2	1188	38	624	10	588	5	432	2
9:15	372	4	480	4	1128	24	576	8	348	3	456	18
9:20	336	4	456	3	816	14	456	6	444	4	540	4
9:25	288	4	432	3	1116	21	456	7	444	3	516	4
9:30	336	3	480	3	1152	20	468	8	528	4	528	4
9:35	300	3	396	2	972	23	276	4	708	6	396	2
9:40	240	2	612	4	1068	20	336	5	420	3	636	4
9:45	324	4	612	4	1068	20	432	6	480	4	468	3
9:50	468	5	576	4	948	17	600	8	480	3	564	4
9:55	384	4	468	3	1080	20	384	6	780	7	444	3
10:00	420	5	420	3	996	17	396	6	432	3	504	3

	detector 857		detector 830		detector 829		detector 896		detector 818		detector 817	
TIME	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
7:00	156	1	1260	9	192	1	60	0	1224	12	300	5
7:05	144	1	1236	9	180	1	60	0	1068	10	312	6
7:10	216	3	1212	9	132	1	84	1	1380	14	324	5
7:15	180	2	1008	7	288	2	156	2	1416	16	276	5
7:20	276	4	1032	7	252	3	168	2	1500	16	384	8
7:25	264	4	1272	9	288	2	108	1	1344	13	552	11
7:30	300	4	948	7	180	1	108	1	1512	15	480	11
7:35	240	3	960	8	300	3	168	2	1368	14	480	11
7:40	276	4	840	6	300	3	192	2	1116	11	624	15
7:45	216	3	1020	8	264	2	216	3	1212	12	564	13
7:50	192	3	1032	8	252	2	312	4	996	9	612	13
7:55	204	3	972	7	240	5	168	2	1272	12	504	13
8:00	192	4	960	7	264	3	156	2	1308	13	600	13
8:05	192	3	1104	8	204	2	180	2	1320	13	348	8
8:10	156	2	984	7	216	2	144	2	1620	18	384	8
8:15	192	3	1188	8	288	2	180	2	1440	14	408	8
8:20	264	2	1212	9	252	2	228	2	1236	12	432	6
8:25	228	2	1176	9	156	1	168	1	1584	15	240	3
8:30	324	3	1308	10	156	1	144	1	1392	14	336	5
8:35	252	2	1356	10	180	1	252	2	1548	19	420	5
8:40	240	2	1512	11	192	1	108	1	1440	15	192	2
8:45	300	2	1440	10	336	3	144	1	1116	10	384	5
8:50	228	2	1392	10	216	2	228	1	1356	14	264	3
8:55	168	1	1212	9	204	2	156	1	1032	10	336	5
9:00	168	1	1284	10	216	2	108	0	1032	10	192	3
9:05	192	2	1308	10	216	2	276	2	1104	11	276	9
9:10	204	2	768	6	348	3	192	1	888	9	312	4
9:15	84	0	900	6	192	1	180	1	924	9	180	2
9:20	180	1	912	6	192	1	204	1	756	7	252	3
9:25	300	2	1020	7	276	2	264	2	972	10	144	2
9:30	156	1	1020	7	180	1	144	1	912	10	264	4
9:35	168	1	984	7	264	2	276	2	684	7	192	2
9:40	204	1	1104	8	228	2	108	0	924	9	288	4
9:45	288	3	804	5	168	1	156	1	840	8	276	4
9:50	216	2	1260	9	264	2	120	0	948	9	144	1
9:55	228	2	1260	9	228	2	216	1	804	7	264	3
10:00	240	2	1152	8	252	2	168	1	996	10	240	3

	detector 807		detector 806		detector 814		detector 1213		detector 1216		detector 1219	
TIME	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
7:00	372	5	96	1	492	2	1488	15	300	2	492	4
7:05	192	3	144	1	516	2	1212	12	276	2	468	4
7:10	360	5	48	1	480	2	1152	11	300	2	552	4
7:15	324	4	144	1	480	2	1236	12	360	2	456	3
7:20	372	5	168	3	516	3	1272	12	348	2	552	4
7:25	288	4	156	2	516	3	1536	15	336	2	444	3
7:30	300	4	180	3	516	3	1488	16	480	3	600	5
7:35	432	6	192	3	444	2	1380	14	444	3	444	3
7:40	444	6	192	3	480	3	1572	16	408	2	576	5
7:45	420	6	192	2	480	3	1488	16	480	3	648	5
7:50	468	7	240	4	396	2	1584	16	300	2	708	6
7:55	444	6	204	3	492	3	1644	18	504	4	504	4
8:00	348	5	288	4	636	3	1584	36	480	3	720	6
8:05	360	5	264	4	624	3	1620	18	372	2	600	5
8:10	480	7	240	3	624	3	1476	17	504	3	504	4
8:15	540	7	216	3	648	3	1512	16	444	4	456	3
8:20	492	8	192	2	576	3	1320	13	480	3	648	5
8:25	480	7	192	1	576	3	1212	12	408	3	612	5
8:30	588	9	180	2	732	4	1380	15	360	2	432	3
8:35	672	10	240	2	408	2	1428	14	444	3	552	4
8:40	492	7	156	1	468	2	1152	11	396	3	432	3
8:45	420	6	168	1	468	2	1128	11	384	2	264	2
8:50	420	6	240	2	612	3	1356	16	564	4	504	4
8:55	504	7	252	2	480	2	1272	13	312	2	456	4
9:00	360	5	240	2	432	2	1212	13	348	3	492	4
9:05	456	7	96	0	408	2	984	10	168	1	444	3
9:10	432	6	180	1	420	2	1116	12	252	2	408	3
9:15	312	4	264	2	384	2	912	9	300	2	408	3
9:20	168	3	180	1	360	2	864	9	384	3	276	2
9:25	192	3	120	1	420	2	780	7	348	2	300	2
9:30	204	3	132	1	504	3	984	9	192	1	288	2
9:35	276	4	96	0	312	1	852	9	324	2	264	2
9:40	252	4	156	1	288	2	864	9	456	3	372	3
9:45	240	4	192	1	336	2	744	7	456	4	408	3
9:50	312	5	216	2	216	1	768	8	252	2	228	1
9:55	348	5	168	1	312	1	1116	12	336	3	252	2
10:00	228	3	360	3	336	1	996	12	324	2	420	3

TIME	detector 1220		detector 1221		detector 1222		detector 1223		detector 1226		detector 1227	
	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
7:00	408	3	48	0	144	2	0	100	0	0	432	4
7:05	444	3	48	0	204	2	0	100	0	0	312	2
7:10	432	4	24	0	180	2	0	100	0	0	480	4
7:15	432	3	60	0	228	2	0	100	0	0	504	4
7:20	312	2	84	1	312	3	0	100	24	0	540	5
7:25	384	3	72	0	324	4	0	100	0	0	600	5
7:30	420	3	132	1	252	3	0	100	12	0	684	5
7:35	516	4	84	1	648	8	0	99	12	0	684	6
7:40	360	2	72	0	396	5	0	100	24	0	612	5
7:45	324	2	24	0	432	5	0	100	24	0	1080	9
7:50	300	2	72	0	624	8	0	100	0	0	648	5
7:55	432	3	36	0	516	6	0	100	12	0	852	8
8:00	468	3	48	0	432	6	0	99	12	0	768	7
8:05	360	2	108	1	336	4	0	100	36	0	876	8
8:10	408	2	24	0	360	4	0	100	12	0	516	4
8:15	480	4	84	0	264	3	0	100	120	1	576	5
8:20	348	2	108	1	432	5	0	100	132	1	600	5
8:25	420	3	108	1	312	3	0	100	24	0	372	3
8:30	228	1	96	1	108	1	0	100	12	0	372	4
8:35	384	3	132	1	180	2	0	100	12	0	432	4
8:40	372	3	48	0	132	2	0	100	0	0	456	4
8:45	360	3	84	1	144	2	0	100	48	0	504	5
8:50	396	3	84	1	132	2	0	100	12	0	324	2
8:55	432	3	36	0	156	2	0	100	0	0	336	3
9:00	360	3	72	1	204	4	0	100	12	0	348	2
9:05	372	4	72	0	108	1	0	100	24	0	252	2
9:10	396	3	48	0	144	2	0	100	12	0	384	4
9:15	372	3	60	0	144	2	0	100	12	0	300	2
9:20	336	3	48	0	216	3	0	100	12	0	372	4
9:25	312	3	72	0	192	3	0	100	0	0	216	2
9:30	324	2	36	0	204	3	0	100	48	0	312	3
9:35	324	2	24	0	216	3	0	100	24	0	180	1
9:40	432	3	72	0	120	1	0	100	12	0	228	2
9:45	492	4	36	0	144	2	0	100	12	0	300	3
9:50	420	4	48	0	204	3	0	100	48	0	156	1
9:55	324	2	0	0	144	2	0	99	12	0	288	2
10:00	300	2	24	0	84	1	0	100	24	0	216	2

	detector 1233		detector 1236		detector 1842		detector 1845		detector 1846		detector 1847	
TIME	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
7:00	0	100	0	100	72	0	0	100	0	100	132	0
7:05	0	100	0	100	108	0	0	100	0	100	108	0
7:10	0	100	0	100	72	0	0	100	0	100	84	0
7:15	0	100	0	100	60	0	0	100	0	100	156	1
7:20	0	100	0	100	144	1	0	100	0	100	120	0
7:25	0	100	0	100	60	0	0	100	0	100	252	1
7:30	0	100	0	100	192	1	0	100	0	100	204	1
7:35	0	100	0	100	120	0	0	100	0	100	84	0
7:40	0	100	0	100	84	0	0	100	0	100	204	1
7:45	0	100	0	100	108	1	0	100	0	100	252	1
7:50	0	100	0	100	144	1	0	100	0	100	324	2
7:55	0	100	0	100	180	1	0	100	0	100	228	1
8:00	0	100	0	100	240	1	0	100	0	100	168	1
8:05	0	100	0	100	72	0	0	100	0	100	156	1
8:10	0	100	0	100	144	1	0	100	0	100	156	1
8:15	0	100	0	100	72	0	0	100	0	100	84	0
8:20	0	100	0	100	144	1	0	100	0	100	132	0
8:25	0	100	0	100	60	0	0	99	0	99	156	1
8:30	0	100	0	100	120	1	0	100	0	100	36	0
8:35	0	99	0	99	48	0	0	100	0	100	180	1
8:40	0	99	0	99	144	1	0	100	0	100	180	1
8:45	0	100	0	100	60	0	0	100	0	100	192	1
8:50	0	100	0	100	108	1	0	99	0	99	108	0
8:55	0	100	0	100	72	0	0	100	0	100	240	1
9:00	0	100	0	100	12	0	0	100	0	100	120	0
9:05	0	100	0	100	108	1	0	100	0	100	96	0
9:10	0	100	0	100	60	0	0	100	0	100	168	1
9:15	0	100	0	100	60	0	0	100	0	100	180	1
9:20	0	100	0	100	48	0	0	100	0	100	96	0
9:25	0	99	0	99	60	0	0	100	0	100	108	0
9:30	0	99	0	99	72	0	0	100	0	100	120	0
9:35	0	100	0	100	108	0	0	100	0	100	132	1
9:40	0	100	0	100	96	0	0	100	0	100	96	0
9:45	0	99	0	99	36	0	0	100	0	100	132	0
9:50	0	100	0	100	60	0	0	100	0	100	84	0
9:55	0	100	0	100	24	0	0	100	0	100	120	0
10:00	0	100	0	100	60	0	0	100	0	100	132	1

	detector 1850		detector 1853		detector 1649		detector 1860		detector 1810	
TIME	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
7:00	192	2	300	3	876	20	216	2	264	3
7:05	252	3	552	6	1056	25	216	2	348	5
7:10	108	1	420	4	1104	32	204	2	336	5
7:15	288	3	492	5	1044	25	156	2	240	3
7:20	288	3	864	10	1080	30	120	1	312	4
7:25	384	4	504	5	936	27	192	2	420	5
7:30	336	4	648	8	948	29	264	3	432	6
7:35	312	3	480	5	888	27	168	2	396	5
7:40	360	4	636	7	876	26	336	4	336	5
7:45	348	3	552	6	924	27	252	3	504	6
7:50	384	4	648	7	960	28	276	4	588	9
7:55	456	4	612	7	972	28	240	3	468	6
8:00	480	5	492	5	984	29	288	3	588	8
8:05	384	4	540	6	984	29	300	4	480	6
8:10	372	4	456	5	960	28	312	3	540	7
8:15	204	2	528	6	972	29	336	4	588	8
8:20	192	2	480	5	984	28	180	2	612	9
8:25	228	3	504	5	1188	33	252	3	504	7
8:30	228	2	336	3	1260	27	300	4	672	9
8:35	144	1	432	5	864	9	252	3	660	9
8:40	156	2	372	4	972	10	180	3	408	6
8:45	204	2	456	5	804	8	240	3	312	5
8:50	180	2	408	4	960	9	300	5	336	4
8:55	228	2	240	2	960	9	168	2	588	9
9:00	192	2	252	3	672	6	144	2	504	6
9:05	132	1	276	3	564	5	216	2	456	6
9:10	120	1	348	4	540	6	132	1	324	4
9:15	120	1	252	3	492	4	120	1	336	5
9:20	96	1	336	3	624	7	132	1	420	6
9:25	156	1	240	3	756	7	144	2	336	5
9:30	132	1	408	5	828	8	180	2	264	3
9:35	96	1	360	4	720	7	108	1	300	5
9:40	144	1	300	4	492	5	96	1	336	4
9:45	84	0	288	4	492	5	240	4	396	6
9:50	228	2	300	3	588	7	168	2	288	4
9:55	96	1	312	4	636	6	180	2	264	4
10:00	120	1	336	4	612	6	84	1	252	3

## Calibrations for I-494 WB South of the Twin Cities International Airport

### FRESIM :

- default vehicle set.
- car following factors = 15 to 6.
- acceleration lag = 0.3 sec.
- deceleration lag = 0.3 sec.
- lane change time = 3 sec.
- vehicle generation minimum separation = 0.5 sec.
- percentage of drivers willing to yield = 30 percent.
- minimum non-emergency deceleration = 0.8 ft/sec<sup>2</sup>

### A IMSUN2 :

#### vehicle characteristics:

- length = 4.0 m.
- width = 1.7 m.
- desired speed = 104 kph.
- maximum acceleration = 2.8 m/sec<sup>2</sup>
- maximum deceleration = 4.0 m/sec<sup>2</sup>

#### simulation parameters:

- simulation step size = 0.75sec
- speed acceptance = 0.80
- braking coefficient = 2.0
- zone 1 distance = 120 m
- zone 2 distance = 40 m
- minimum distance between vehicles = 1.2 m
- queue speed = 1.0 m/sec<sup>2</sup>
- queue leaving speed = 4.0 m/sec<sup>2</sup>
- junction visibility = 25 m
- cruising tolerance = 0.80

### FREFLO :

- relaxation time = 75 sec/mile
- anticipation coefficient = 0.25 miles<sup>2</sup>/hour
- capacities as listed on the freeway geometrics
- cubic polynomial speed-density relationship

$$\text{speed} = 106 - 160(k * 10^{-2}) + 76(k * 10^{-2})^2 - 7(k * 10^{-2})^3$$

FREQ11:

- capacities as listed on the freeway geometrics
- points on the speed-V/C curve

Points on Upper Portion		of Speed-V/C curve	Points on Lower Portion		of Speed-V/C curve
V/C ratio	speed (mph)		V/C ratio	speed (mph)	
0.00	65		0.00	0	
0.72	65		0.39	11	
0.93	61		0.81	26	
1.00	53		1.00	53	

KRONOS:

- jam density = 142 vpm
- maximum capacity = 2300 vph
- density at maximum capacity = 46 vpm
- density at end of linear region of QK curve = 22 vpm
- flow rate at end of linear region of QK curve = 1450 vph
- capacities as listed on the freeway geometrics

INTEGRATION:

- mainline freeflow speed = 104 kph
- ramp freeflow speed = 88 kph
- capacities as listed on the freeway geometrics
- speed at capacity for mainline and 2100 vph capacity ramps = 72 kph
- speed at capacity for all other ramps = 56.4 kph
- mainline jam density when capacity is 2400 vphpl = 91.7 vpkpl
- mainline jam density when capacity is 2200 vpkpl = 85.1 vpkpl
- ramp jam densities = 130 vpkpl



## Calibrations For I-494 WB Through Southwest Minneapolis

### FRESIM :

- default vehicle set.
- car following factors = 15 to 6.
- acceleration lag = 0.3 sec.
- deceleration lag = 0.3 sec.
- lane change time = 3 sec.
- vehicle generation minimum separation = 1.0 sec.
- percentage of drivers willing to yield = 30 percent.
- minimum non-emergency deceleration = 0.8 ft/sec<sup>2</sup>

### AIMSUN2 :

#### vehicle characteristics:

- length = 4.0 m.
- width = 1.7 m.
- desired speed = 104 kph.
- maximum acceleration = 2.8 m/sec<sup>2</sup>
- maximum deceleration = 4.0 m/sec<sup>2</sup>

#### simulation parameters:

- simulation step size = 0.75sec
- speed acceptance = 0.80
- braking coefficient = 2.0
- zone 1 distance = 120 m
- zone 2 distance = 40 m
- minimum distance between vehicles = 1.2 m
- queue speed = 1.0 m/sec<sup>2</sup>
- queue leaving speed = 4.0 m/sec<sup>2</sup>
- junction visibility = 25 m
- cruising tolerance = 0.80

### FREFLO :

- relaxation time = 75 sec/mile
- anticipation coefficient = 0.25 miles<sup>2</sup>/hour
- capacities as listed on the freeway geometrics
- cubic polynomial speed-density relationship

$$\text{speed} = 87 - 49 (k * 10^{-2}) - 100 (k * 10^{-2})^2 + 64 (k * 10^{-2})^3$$

FREQ11:

- capacities as listed on the freeway geometrics
- points on the speed-V/C curve

Points on Upper of Speed-V/C curve Portion		Points on Lower of Speed-V/C curve Portion	
V/C ratio	speed (mph)	V/C ratio	speed (mph)
0.00	65	0.00	0
0.71	65	0.34	9
0.92	44	0.78	21
1.00	34	1.00	34

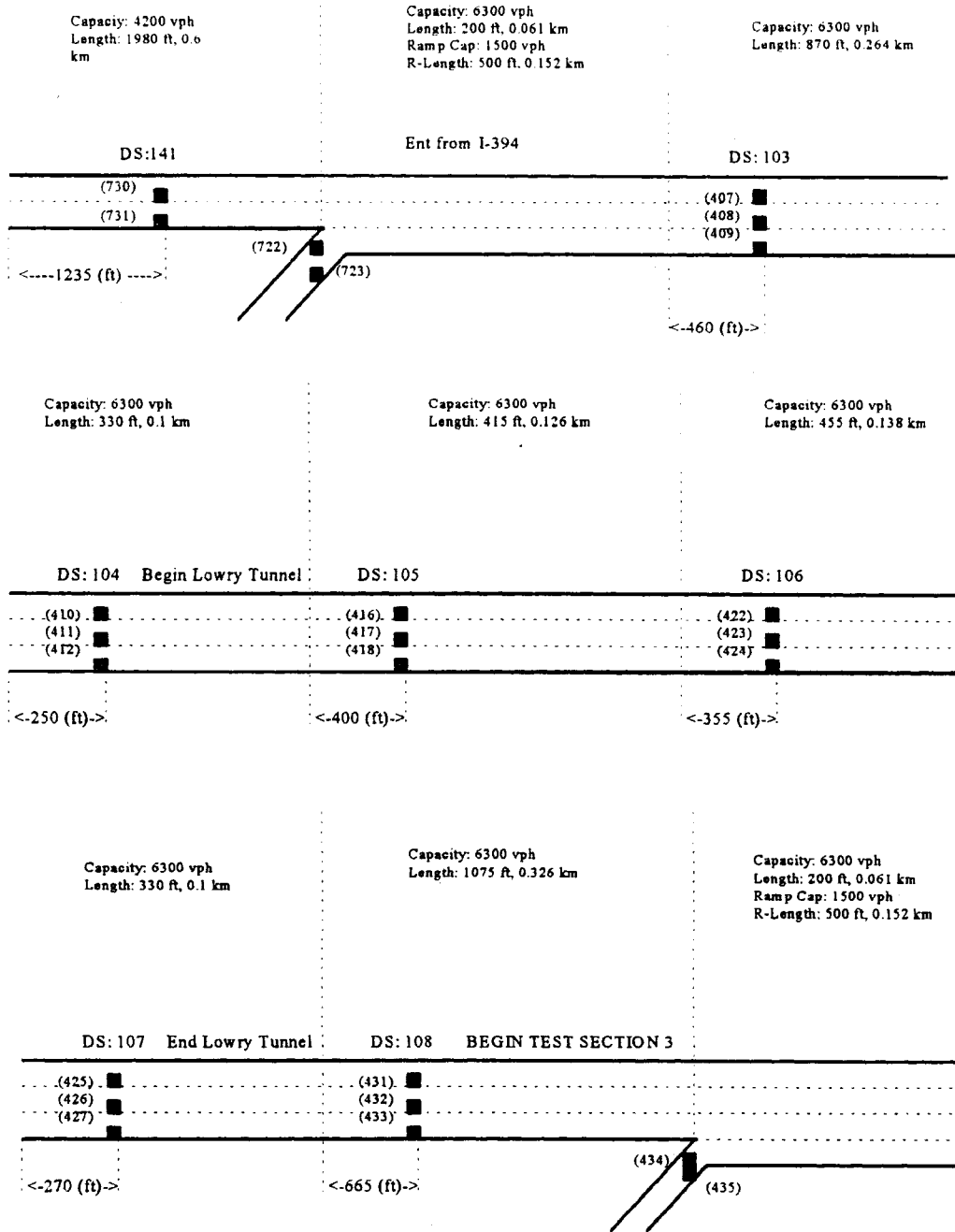
KRONOS:

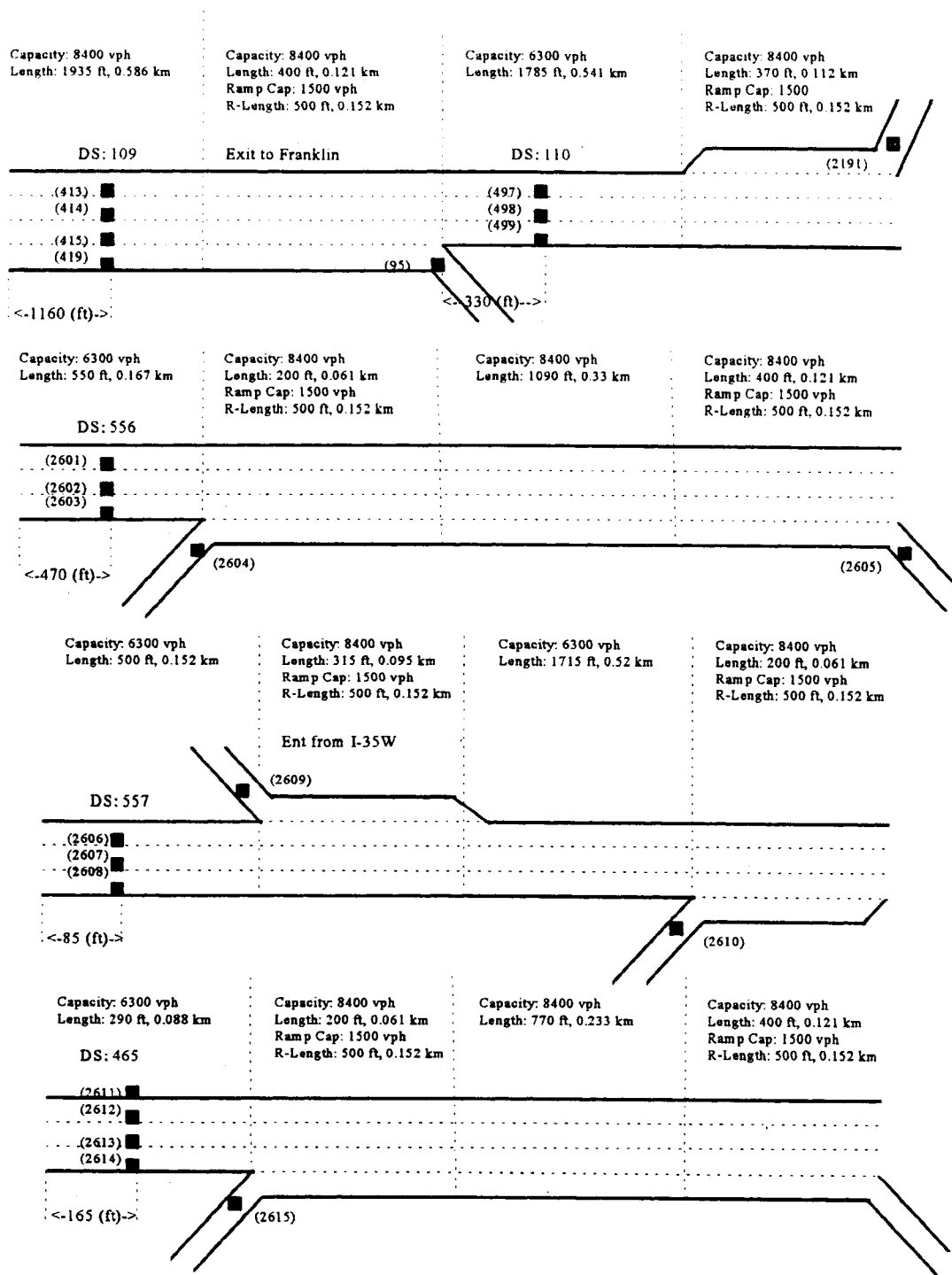
- jam density = 136 vpm
- maximum capacity = 2200 vph
- density at maximum capacity = 41 vpm
- density at end of linear region of QK curve = 24 vpm
- flow rate at end of linear region of QK curve = 1580 vph
- capacities as listed on the freeway geometrics

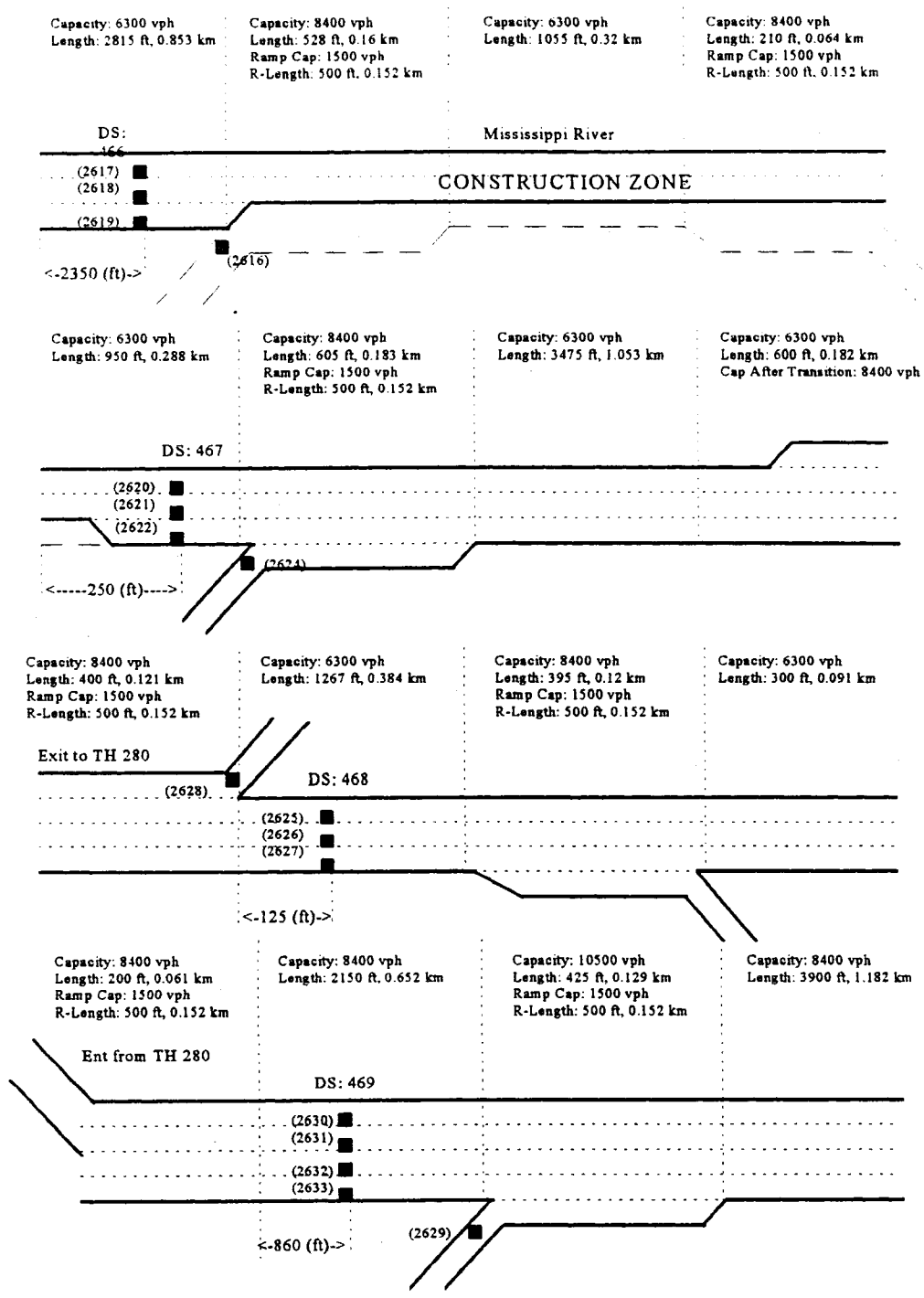
INTEGRATION:

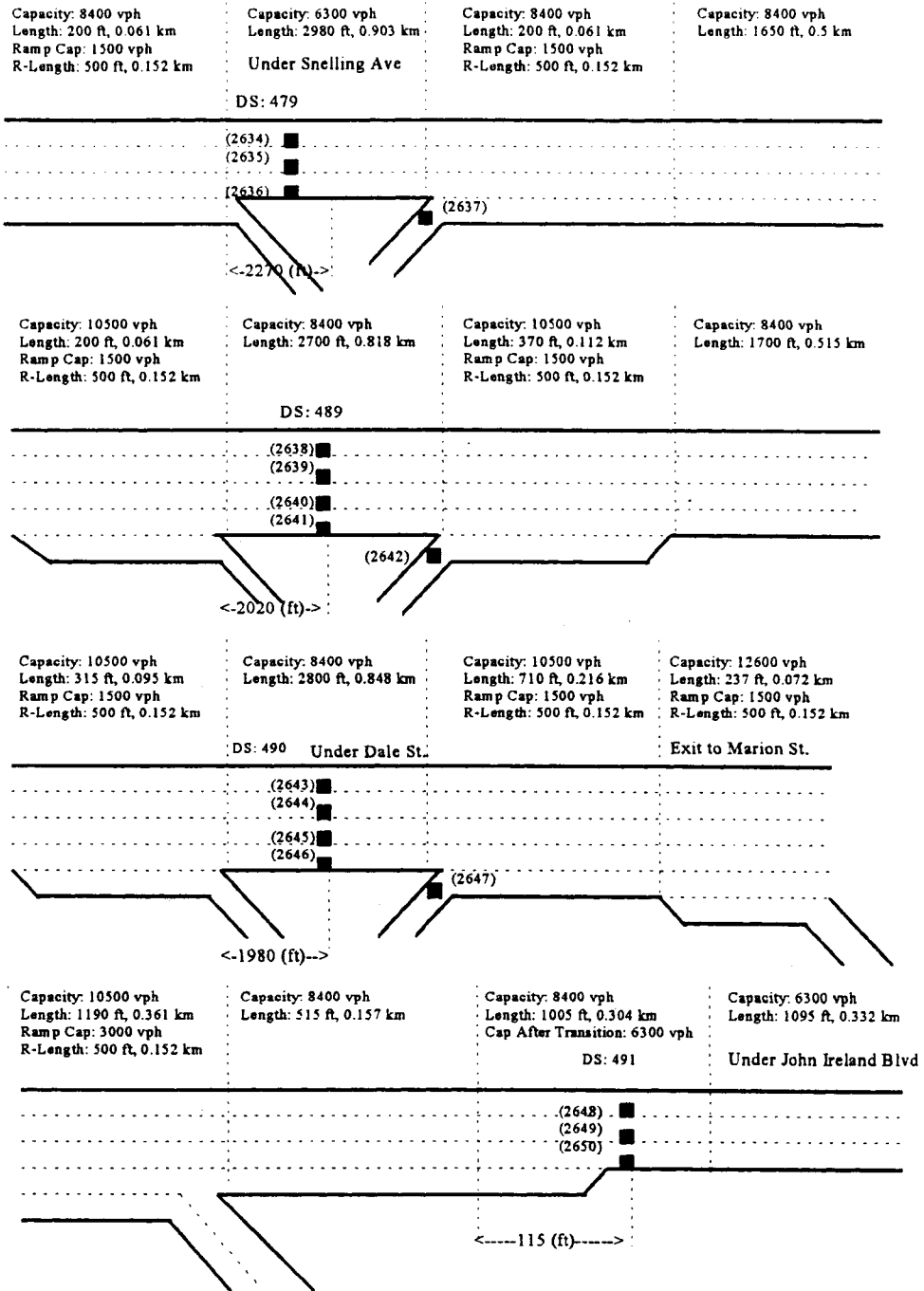
- mainline freeflow speed = 104 kph
- ramp freeflow speed for capacities less than 2100 vph = 89 kph
- ramp freeflow speed when capacity is 2100 vph = 104 kph
- capacities as listed on the freeway geometrics
- speed at capacity for mainline and 2100 vph capacity ramps = 72 kph
- speed at capacity for all other ramps = 56.4 kph
- mainline jam density when capacity is 2400 vphpl = 91.7 vpkpl
- mainline jam density when capacity is 2300 vphpl = 87.9 vpkpl
- mainline jam density when capacity is 2200 vpkpl = 85.1 vpkpl
- ramp jam density when capacity is 2100 vph = 80.3 vpkpl
- all other ramp jam densities = 130 vpkpl

## TEST FREEWAY SECTION (I-94 Eastbound, Twin Cities)









DETECTOR STATION 141

DETECTOR STATION 103

TIME	detector 730		detector 731	
	VOL	OCC	VOL	OCC
6:55	1836	21	1944	18
7:00	1716	17	1500	11
7:05	1452	13	1548	11
7:10	1644	14	1620	12
7:15	1572	13	1536	11
7:20	1752	22	1908	16
7:25	1836	30	1908	22
7:30	1752	27	1848	19
7:35	1956	20	1620	13
7:40	1764	31	1848	21
7:45	1524	34	1920	29
7:50	1752	28	1896	22
7:55	1644	36	1968	29
8:00	1548	29	1848	23
8:05	1680	15	1548	11
8:10	1572	15	1512	11
8:15	1416	11	1248	8
8:20	1812	17	1716	13
8:25	1740	15	1620	11
8:30	1716	17	1620	12
8:35	1656	16	1512	11
8:40	1560	17	1380	10
8:45	1620	15	1356	9
8:50	1548	15	1296	9
8:55	1356	11	1164	8
9:00	1560	15	1488	10
9:05	1416	12	1032	7
9:10	1140	10	1008	7
9:15	1212	9	924	6
9:20	1308	11	876	6
9:25	1296	13	1068	8
9:30	1044	10	804	5
9:35	1272	11	816	5
9:40	1200	10	1032	7
9:45	1368	13	1068	7
9:50	1428	14	1092	7
9:55	1308	12	1020	7
10:00	1284	11	852	6

detector 407		detector 408		detector 409	
VOL	OCC	VOL	OCC	VOL	OCC
2340	27	1980	22	1860	18
2484	26	1896	20	1656	15
1992	21	1956	21	1728	14
2208	22	2124	22	1836	16
2136	23	2208	23	1692	14
2172	29	2160	28	1968	19
2100	31	2160	31	1968	21
2268	36	1944	30	1920	21
2196	30	2208	29	1896	20
2220	33	1980	27	1884	20
2136	29	2004	29	2124	24
2148	36	1920	29	2088	24
2088	34	2016	32	2184	28
2088	29	1968	28	2112	25
2160	30	1968	25	1860	20
2208	25	2088	22	1596	14
2064	20	1764	16	1404	12
2292	26	1932	21	1800	16
2040	22	2100	21	1800	16
2292	22	1896	18	1680	15
2004	25	2004	24	1716	16
2112	26	1848	22	1800	18
2160	21	1932	19	1440	12
2124	22	1980	21	1440	12
1956	19	1860	18	1368	11
1728	18	1872	21	1572	15
1668	18	1692	19	1344	12
1740	16	1572	15	1068	9
1740	17	1728	18	1068	10
1788	18	1620	16	1020	8
1716	19	1776	20	1176	11
1788	22	1644	18	1068	10
1848	20	1716	18	936	7
1920	18	1800	18	1128	9
1980	20	1812	20	1248	10
1848	23	1908	23	1296	12
1872	18	1764	19	1284	10
1776	17	1608	15	1164	9

DETECTOR STATION 104

TIME	detector 410		detector 411		detector 412	
	VOL	OCC	VOL	OCC	VOL	OCC
6:55	2124	25	1860	19	1980	19
7:00	2052	22	1836	18	1860	17
7:05	1800	18	1908	19	1896	17
7:10	2004	20	1968	20	2088	19
7:15	2028	22	1956	20	2016	18
7:20	1980	25	1992	22	2148	21
7:25	1968	26	1932	23	2232	24
7:30	2064	28	1848	23	2148	24
7:35	2040	26	1980	25	2148	25
7:40	2064	25	1980	24	2004	22
7:45	1944	25	1884	25	2268	26
7:50	2004	29	1944	26	2148	25
7:55	1896	27	2016	29	2256	33
8:00	1980	26	1884	23	2184	27
8:05	1980	27	1752	23	1992	23
8:10	1812	21	2088	24	1824	16
8:15	1596	16	1740	17	1620	14
8:20	2088	24	1836	19	1908	17
8:25	1824	20	1932	19	2064	19
8:30	1836	20	1716	18	1908	17
8:35	1764	22	1872	22	1896	18
8:40	1908	23	1764	21	2004	20
8:45	1884	20	1740	18	1620	15
8:50	1848	20	1980	20	1560	14
8:55	1716	18	1728	17	1584	13
9:00	1572	17	1644	19	1752	17
9:05	1428	16	1596	18	1608	15
9:10	1332	13	1524	15	1212	11
9:15	1488	16	1560	16	1344	13
9:20	1416	15	1596	16	1296	11
9:25	1452	15	1632	19	1416	13
9:30	1632	21	1428	17	1212	11
9:35	1560	18	1512	17	1308	11
9:40	1704	17	1536	16	1368	12
9:45	1608	17	1704	18	1428	12
9:50	1620	21	1740	21	1596	16
9:55	1692	18	1608	18	1608	14
10:00	1440	15	1620	16	1380	11

DETECTOR STATION 105

detector 416		detector 417		detector 418	
VOL	OCC	VOL	OCC	VOL	OCC
0	0	1896	19	1908	20
0	0	1800	19	1956	20
0	0	1692	18	1968	20
0	0	1740	19	2136	22
0	0	1776	18	2076	21
0	0	1992	22	2136	23
0	0	1836	22	2136	25
0	0	1800	23	2220	27
0	0	1932	25	2148	27
0	0	1956	25	1920	23
0	0	1872	25	2148	27
0	0	1836	25	2112	27
0	0	1896	26	2148	28
0	0	1908	25	2088	30
0	0	1764	24	2124	28
0	0	1800	21	1992	21
0	0	1644	17	1668	16
0	0	1752	19	1872	20
0	0	1668	18	2160	23
0	0	1776	20	1860	20
0	0	1740	20	1920	21
0	0	1584	20	2100	23
0	0	1668	18	1584	16
0	0	1692	19	1752	18
0	0	1572	17	1620	16
0	0	1608	19	1704	18
0	0	1476	18	1728	18
0	0	1380	14	1332	13
0	0	1440	15	1296	14
0	0	1416	16	1464	15
0	0	1524	19	1428	15
0	0	1392	17	1368	15
0	0	1488	18	1404	14
0	0	1440	16	1452	15
0	0	1500	19	1536	16
0	0	1608	22	1644	19
0	0	1428	18	1728	18
0	0	1452	16	1536	14



DETECTOR STATION 106

TIME	detector 422		detector 423		detector 424	
	VOL	OCC	VOL	OCC	VOL	OCC
6:55	2088	23	1728	17	1968	20
7:00	1956	21	1764	18	1980	20
7:05	1812	19	1668	16	1956	20
7:10	1896	19	1680	17	2076	21
7:15	1968	21	1740	17	2088	21
7:20	1992	26	1824	18	2028	21
7:25	1920	23	1848	19	2148	23
7:30	2124	26	1656	18	2196	24
7:35	2016	23	1764	19	2184	25
7:40	2076	24	1836	20	1992	22
7:45	1944	24	1812	20	2160	25
7:50	2040	26	1740	20	2124	24
7:55	2004	24	1860	21	2088	24
8:00	1872	23	1920	21	2004	24
8:05	1992	24	1764	21	2160	24
8:10	1752	20	1692	19	1968	21
8:15	1548	17	1656	16	1752	17
8:20	2052	25	1584	16	1968	19
8:25	1776	20	1632	16	2124	22
8:30	1776	20	1704	18	1992	21
8:35	1692	20	1644	17	1992	21
8:40	1824	24	1620	18	2124	23
8:45	1908	22	1512	16	1584	16
8:50	1752	21	1620	17	1800	19
8:55	1668	19	1704	17	1644	16
9:00	1584	18	1524	16	1668	18
9:05	1392	17	1464	16	1728	17
9:10	1212	13	1296	13	1344	14
9:15	1464	17	1392	14	1332	14
9:20	1284	15	1356	14	1524	15
9:25	1404	15	1464	17	1440	14
9:30	1452	20	1296	15	1452	15
9:35	1476	17	1368	16	1500	15
9:40	1572	17	1440	16	1500	15
9:45	1464	17	1452	17	1668	16
9:50	1584	20	1512	19	1788	20
9:55	1572	18	1344	16	1764	19
10:00	1368	15	1404	15	1488	14

DETECTOR STATION 107

detector 425		detector 426		detector 427	
VOL	OCC	VOL	OCC	VOL	OCC
2052	23	1728	18	1956	20
2004	22	1776	19	1944	20
1824	19	1632	17	1932	20
1908	20	1668	18	2028	21
2016	22	1668	17	2076	21
2016	25	1824	19	2028	21
1896	23	1836	19	2112	22
2160	26	1596	18	2148	23
2064	23	1776	19	2172	24
2028	23	1788	20	1956	22
1968	24	1740	20	2124	24
2052	25	1788	21	2088	23
2088	24	1848	21	1992	22
1920	23	1956	21	1980	22
2028	23	1788	20	2172	24
1812	21	1680	20	1908	20
1584	17	1644	17	1764	17
2040	24	1584	17	1956	20
1776	20	1668	17	2064	22
1812	20	1716	19	1932	21
1752	20	1680	18	1932	20
1812	23	1608	19	2076	23
1896	21	1560	16	1572	17
1800	21	1608	18	1776	19
1716	20	1692	18	1596	16
1584	18	1524	17	1644	18
1440	18	1476	17	1704	18
1224	13	1260	13	1332	14
1440	17	1404	15	1344	14
1260	15	1344	14	1476	16
1404	15	1476	19	1416	15
1488	20	1236	15	1452	16
1500	18	1332	16	1452	15
1608	18	1440	16	1416	15
1524	18	1428	18	1608	17
1572	20	1548	20	1752	20
1596	18	1380	18	1716	19
1356	16	1404	16	1524	16

DETECTOR STATION 108

TIME	detector 431		detector 432		detector 433	
	VOL	OCC	VOL	OCC	VOL	OCC
6:55	1968	18	1752	17	2064	21
7:00	1872	17	1572	15	1848	18
7:05	1872	17	1572	15	1848	18
7:10	2076	19	1548	14	2064	20
7:15	2052	18	1656	15	1980	20
7:20	2100	19	1680	15	2064	22
7:25	2160	20	1584	15	2256	25
7:30	2124	21	1716	16	2172	22
7:35	1848	17	1776	17	2088	21
7:40	1848	17	1776	17	2088	21
7:45	2172	19	1680	18	2076	24
7:50	2052	18	1668	17	2208	26
7:55	1956	18	1764	17	2088	21
8:00	1956	18	1764	17	2088	21
8:05	2184	21	1800	18	2124	23
8:10	1848	20	1656	19	1896	22
8:15	1920	16	1572	15	2112	22
8:20	1920	16	1572	15	2112	22
8:25	2028	18	1656	15	1848	19
8:30	1968	17	1596	14	1788	18
8:35	1968	17	1596	14	1788	18
8:40	2100	19	1632	16	1884	21
8:45	1500	13	1428	13	1956	19
8:50	1716	15	1596	16	1860	19
8:55	1548	13	1704	15	1752	18
9:00	1644	15	1476	15	1596	16
9:05	1704	16	1356	15	1548	17
9:10	1416	13	1212	11	1284	13
9:15	1320	12	1308	13	1452	16
9:20	1452	13	1320	12	1332	14
9:25	1488	13	1452	16	1380	13
9:30	1440	13	1368	15	1440	17
9:35	1332	12	1380	14	1464	15
9:40	1416	13	1452	15	1668	17
9:45	1548	13	1428	15	1512	16
9:50	1668	15	1416	16	1728	19
9:55	1728	16	1224	14	1680	18
10:00	1476	12	1440	15	1404	14

DETECTOR STATION 109

detector 413		detector 414	
VOL	OCC	VOL	OCC
1788	16	1524	13
1296	10	1452	11
1596	13	1260	10
1584	13	1548	12
1692	16	1416	12
1896	17	1296	11
1944	18	1596	13
1608	14	1728	15
1800	16	1632	14
1716	16	1728	16
1860	17	1620	15
2124	31	1536	15
1860	20	1716	20
1716	17	1740	21
1752	19	1728	20
1500	14	1692	16
1812	16	1536	12
1560	13	1560	12
1752	14	1260	9
1548	13	1308	10
1620	15	1392	11
1404	11	1428	10
1416	12	1596	13
1548	14	1332	11
1500	13	1332	10
1320	11	1152	9
1320	11	1152	9
1080	9	1092	9
1140	10	1092	10
1128	10	1272	10
1044	9	1188	10
1128	10	1200	10
1104	10	1248	10
1200	10	1248	10
1188	11	1164	9
1440	12	1200	9
1320	12	1320	11
972	8	1212	10

DETECTOR STATION 109

TIME	detector 415		detector 419	
	VOL	OCC	VOL	OCC
6:55	0	0	2112	17
7:00	0	0	2100	15
7:05	0	0	2004	16
7:10	0	0	2304	17
7:15	0	0	2220	17
7:20	0	0	2436	19
7:25	0	0	2280	17
7:30	0	0	2244	19
7:35	0	0	2292	20
7:40	0	0	2232	21
7:45	0	0	2388	25
7:50	0	0	2208	24
7:55	0	0	1812	36
8:00	0	0	1668	38
8:05	0	0	1572	25
8:10	0	0	1680	32
8:15	0	0	2040	15
8:20	0	0	2220	16
8:25	0	0	1944	14
8:30	0	0	2064	15
8:35	0	0	2388	18
8:40	0	0	1764	13
8:45	0	0	2100	16
8:50	0	0	1848	14
8:55	0	0	1980	16
9:00	0	0	1848	14
9:05	0	0	1848	14
9:10	0	0	1704	13
9:15	0	0	1428	11
9:20	0	0	1596	13
9:25	0	0	1644	13
9:30	0	0	1596	13
9:35	0	0	1596	12
9:40	0	0	1680	13
9:45	0	0	1680	12
9:50	0	0	1848	14
9:55	0	0	1884	15
10:00	0	0	1488	12

DETECTOR STATION 110

detector 497		detector 498		detector 499	
VOL	OCC	VOL	OCC	VOL	OCC
1680	14	1320	9	2004	16
1428	12	1572	11	2196	19
1404	11	1548	11	2208	18
1236	9	1440	11	2172	18
1524	12	1536	11	2352	19
1284	10	1680	13	2376	20
1452	12	1728	13	2460	23
1536	12	1680	13	2412	21
1632	13	1704	14	2220	23
1716	17	1752	16	2184	25
1884	16	1608	14	2244	23
1728	15	1920	17	2340	27
1680	15	1620	16	2112	29
1968	19	1584	16	1908	40
1680	16	1584	15	2232	28
1488	14	1500	16	1920	35
1836	16	1704	16	2136	26
1368	10	1548	11	2184	17
1524	10	1488	11	2280	17
1272	9	1404	10	2232	18
1248	9	1488	10	2160	16
1308	10	1596	11	2364	18
1344	9	1356	10	2040	15
1308	10	1500	11	2172	17
1152	9	1452	10	2052	16
1356	11	1368	10	2208	18
1056	8	1284	10	1944	15
1140	9	1164	9	1884	15
1116	9	1320	11	1584	13
1188	9	1128	8	1680	13
1164	9	1272	11	1788	14
996	8	1308	10	1812	15
1236	10	1200	10	1728	13
1152	9	1212	10	1812	13
1224	9	1344	11	1716	12
1092	8	1368	12	1884	14
1320	11	1272	11	1908	15
1116	9	1236	9	1956	14

DETECTOR STATION 556

TIME	detector 2601		detector 2602		detector 2603	
	VOL	OCC	VOL	OCC	VOL	OCC
6:55	1488	12	1356	10	684	5
7:00	1404	11	1332	10	792	5
7:05	1164	9	1308	10	660	5
7:10	1440	11	1392	10	684	5
7:15	1224	10	1584	12	936	6
7:20	1284	11	1620	12	732	5
7:25	1440	11	1512	11	984	7
7:30	1512	11	1488	11	708	4
7:35	1548	13	1536	12	732	5
7:40	1632	13	1428	11	732	5
7:45	1632	14	1536	14	924	7
7:50	1392	11	1380	12	684	6
7:55	1788	15	1236	10	672	5
8:00	1500	12	1284	9	624	5
8:05	1368	12	1080	9	552	4
8:10	1632	13	1404	11	696	5
8:15	1368	10	1500	11	696	4
8:20	1392	10	1428	10	528	3
8:25	1260	9	1224	9	480	3
8:30	1092	8	1452	10	648	4
8:35	1308	10	1488	11	768	5
8:40	1224	8	1296	10	612	4
8:45	1272	10	1308	10	552	3
8:50	1128	8	1260	9	420	2
8:55	1224	10	1188	8	480	3
9:00	1044	8	1200	11	600	4
9:05	1116	8	1044	8	348	2
9:10	1116	9	1152	9	456	3
9:15	1032	8	1200	9	432	3
9:20	1104	8	1200	10	504	3
9:25	876	7	1212	10	444	3
9:30	1272	10	1068	8	528	3
9:35	1152	9	1152	10	444	3
9:40	1236	9	1164	10	540	3
9:45	924	7	1308	11	420	2
9:50	1104	9	1260	11	564	4
9:55	1212	9	1128	9	564	3
10:00	1032	8	1092	10	396	2

DETECTOR STATION 557

TIME	detector 2606		detector 2607		detector 2608	
	VOL	OCC	VOL	OCC	VOL	OCC
6:55	828	7	1104	8	924	6
7:00	696	6	1224	9	1080	7
7:05	600	4	1200	8	840	5
7:10	636	5	1176	8	1008	5
7:15	612	4	1344	9	1020	5
7:20	816	6	1440	9	924	5
7:25	720	5	1332	9	1104	7
7:30	828	6	1284	8	924	5
7:35	936	7	1260	8	1032	6
7:40	924	7	1308	8	984	5
7:45	768	6	1284	9	936	4
7:50	744	6	1248	9	888	4
7:55	960	7	1200	8	888	4
8:00	792	6	1272	8	636	3
8:05	744	6	984	6	708	3
8:10	996	7	1200	8	792	4
8:15	876	6	1188	8	948	4
8:20	804	6	1224	8	708	3
8:25	816	5	1104	7	636	3
8:30	684	4	1296	8	756	4
8:35	768	6	1224	8	924	5
8:40	852	6	1320	8	768	4
8:45	756	5	1140	7	708	3
8:50	624	4	1128	7	684	3
8:55	696	5	1056	7	684	3
9:00	648	5	1020	8	708	4
9:05	600	4	972	7	684	4
9:10	660	5	1044	7	588	4
9:15	588	5	1092	7	696	3
9:20	636	4	1188	8	516	2
9:25	576	4	1128	8	636	4
9:30	708	5	936	7	612	4
9:35	708	5	1068	7	588	3
9:40	780	6	1236	9	696	4
9:45	600	5	1188	9	696	4
9:50	516	4	1152	8	600	3
9:55	708	5	996	7	708	4
10:00	600	4	912	7	696	4

DETECTOR STATION 465

TIME	detector 2611		detector 2612		detector 2613		detector 2614	
	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
6:55	648	5	816	5	0	100	1380	12
7:00	924	7	792	5	0	100	1428	11
7:05	672	5	960	7	0	100	1656	15
7:10	420	2	900	6	0	100	1404	11
7:15	312	2	840	6	0	100	1524	12
7:20	312	2	840	7	0	100	1716	13
7:25	372	3	1116	10	0	100	1740	14
7:30	456	3	960	7	0	100	1860	16
7:35	588	5	924	7	0	100	1632	13
7:40	456	3	1236	9	0	100	1656	13
7:45	528	4	996	7	0	100	1692	14
7:50	372	3	1140	9	0	100	1860	15
7:55	480	4	948	7	0	100	1560	12
8:00	528	4	1044	7	0	100	1644	12
8:05	684	5	1104	8	0	100	1416	11
8:10	528	4	1092	8	0	100	1308	10
8:15	684	5	1104	7	0	100	1524	12
8:20	372	3	1212	9	0	100	1608	12
8:25	648	4	1272	9	0	100	1272	9
8:30	420	2	1224	8	0	100	1332	10
8:35	384	3	1032	7	0	100	1428	11
8:40	468	3	1020	8	0	100	1296	10
8:45	516	4	1104	8	0	100	1476	11
8:50	408	2	1068	7	0	100	1356	10
8:55	372	2	1008	6	0	100	1212	9
9:00	372	2	1212	9	0	100	1188	9
9:05	456	3	948	7	0	100	1392	12
9:10	384	2	1032	7	0	100	1236	9
9:15	336	2	936	7	0	100	1152	9
9:20	384	3	1140	10	0	100	1272	10
9:25	216	1	1056	8	0	100	1176	9
9:30	288	2	1068	9	0	100	1428	12
9:35	432	3	996	8	0	100	1056	9
9:40	324	2	1236	9	0	100	1356	11
9:45	432	3	1368	10	0	100	1248	11
9:50	384	2	1152	9	0	100	1404	12
9:55	264	1	972	8	0	100	1272	10
10:00	480	3	1392	11	0	100	1236	11

DETECTOR STATION 466

TIME	detector 2617		detector 2618		detector 2619	
	VOL	OCC	VOL	OCC	VOL	OCC
6:55	0	0	0	0	0	0
7:00	0	0	0	0	0	0
7:05	0	0	0	0	0	0
7:10	0	0	0	0	0	0
7:15	0	0	0	0	0	0
7:20	0	0	0	0	0	0
7:25	0	0	0	0	0	0
7:30	0	0	0	0	0	0
7:35	0	0	0	0	0	0
7:40	0	0	0	0	0	0
7:45	0	0	0	0	0	0
7:50	0	0	0	0	0	0
7:55	0	0	0	0	0	0
8:00	0	0	0	0	0	0
8:05	0	0	0	0	0	0
8:10	0	0	0	0	0	0
8:15	0	0	0	0	0	0
8:20	0	0	0	0	0	0
8:25	0	0	0	0	0	0
8:30	0	0	0	0	0	0
8:35	0	0	0	0	0	0
8:40	0	0	0	0	0	0
8:45	0	0	0	0	0	0
8:50	0	0	0	0	0	0
8:55	0	0	0	0	0	0
9:00	0	0	0	0	0	0
9:05	0	0	0	0	0	0
9:10	0	0	0	0	0	0
9:15	0	0	0	0	0	0
9:20	0	0	0	0	0	0
9:25	0	0	0	0	0	0
9:30	0	0	0	0	0	0
9:35	0	0	0	0	0	0
9:40	0	0	0	0	0	0
9:45	0	0	0	0	0	0
9:50	0	0	0	0	0	0
9:55	0	0	0	0	0	0
10:00	0	0	0	0	0	0

DETECTOR STATION 467

detector 2620		detector 2621		detector 2622	
VOL	OCC	VOL	OCC	VOL	OCC
384	3	1092	9	1560	11
324	2	1092	9	1320	10
288	2	1320	10	1560	13
300	2	1260	9	1704	14
324	2	1428	11	1788	14
276	2	1476	12	1932	14
312	2	1500	13	2016	15
216	2	1620	13	2136	16
336	3	1524	13	2052	16
348	2	1512	11	2124	16
432	3	1476	12	2028	15
432	3	1548	13	2088	16
468	3	1464	12	2172	18
408	2	1308	10	1824	13
360	3	1344	10	1896	13
324	2	1296	10	1884	13
396	3	1176	9	1668	12
288	2	1380	11	2064	15
396	3	1380	11	1968	14
312	2	1296	10	1836	13
324	2	1068	8	1548	12
324	2	1464	12	2016	14
312	2	1404	11	1872	14
252	2	1296	10	1668	12
216	1	1452	11	1836	13
204	2	1116	8	1548	11
300	2	1344	11	1476	11
276	2	1368	12	1728	14
144	1	1296	10	1524	11
336	3	1332	12	1776	14
180	1	1428	12	1776	13
168	1	1668	15	1788	15
180	1	1296	12	1692	15
228	2	1296	11	1488	11
288	2	1440	12	1848	15
372	3	1368	13	1932	16
432	3	1296	11	1752	15
528	5	1296	12	1596	14

DETECTOR STATION 468

TIME	detector 2625		detector 2626		detector 2627	
	VOL	OCC	VOL	OCC	VOL	OCC
6:55	912	7	1032	3	756	6
7:00	648	4	1032	3	672	5
7:05	744	5	1176	3	816	6
7:10	876	6	1248	4	672	5
7:15	852	6	1248	4	732	5
7:20	1044	7	1380	4	912	6
7:25	1092	8	1332	4	840	6
7:30	1032	7	1572	5	780	6
7:35	1296	9	1308	4	936	7
7:40	996	7	1320	4	972	7
7:45	936	6	1452	4	1020	7
7:50	1068	7	1452	4	1080	8
7:55	1044	7	1416	4	1224	9
8:00	888	6	1236	4	984	7
8:05	780	5	1212	4	852	6
8:10	804	5	1152	3	912	7
8:15	708	5	1104	3	852	7
8:20	960	6	1428	4	768	5
8:25	1044	7	1392	4	876	6
8:30	804	5	1212	4	984	7
8:35	588	4	1080	3	720	5
8:40	840	5	1344	4	936	6
8:45	660	5	1428	4	828	6
8:50	732	5	1188	4	936	7
8:55	672	4	1260	4	852	6
9:00	756	5	1140	3	852	6
9:05	648	4	1236	4	756	6
9:10	720	5	1272	4	924	6
9:15	648	4	1140	3	840	6
9:20	624	4	1164	3	804	7
9:25	708	5	1320	4	1020	8
9:30	576	4	1344	4	912	7
9:35	660	5	1368	4	900	7
9:40	456	3	996	3	900	7
9:45	708	5	1212	4	1080	8
9:50	552	4	1308	4	996	8
9:55	540	3	1248	4	1092	8
10:00	672	5	1200	4	1284	11

DETECTOR STATION 469

detector 2630		detector 2631	
VOL	OCC	VOL	OCC
480	3	1104	8
456	3	1092	8
456	3	1092	8
420	3	1008	7
432	3	1248	9
564	4	1344	9
588	4	1200	9
588	4	1452	10
564	4	1308	10
600	4	1344	9
732	5	1356	10
732	5	1356	10
588	4	1404	10
612	5	1224	9
624	5	1344	10
624	5	1236	9
600	5	1140	8
528	4	1308	9
660	5	1308	9
564	4	1296	9
528	4	1092	8
672	5	1164	9
720	5	1284	9
612	4	1224	9
600	5	1236	9
552	4	1056	7
588	4	1152	8
684	5	1140	10
600	4	1140	8
744	6	1128	8
648	5	1224	9
672	5	1260	9
708	5	1272	10
576	4	912	7
648	5	1164	9
612	5	1200	9
696	5	1080	8
684	5	1092	9

DETECTOR STATION 469

TIME	detector 2632		detector 2633	
	VOL	OCC	VOL	OCC
6:55	984	7	708	5
7:00	1008	7	588	4
7:05	1008	7	588	4
7:10	1152	7	624	5
7:15	1176	7	852	6
7:20	1224	8	852	6
7:25	1140	7	876	6
7:30	1344	8	960	7
7:35	1524	10	1068	7
7:40	1308	8	804	5
7:45	1332	8	972	7
7:50	1332	8	972	7
7:55	1332	9	1152	8
8:00	1260	8	900	6
8:05	1164	8	984	7
8:10	1104	7	804	5
8:15	1212	8	852	6
8:20	1188	8	756	6
8:25	1272	8	864	6
8:30	1140	7	840	6
8:35	900	6	648	4
8:40	1188	7	696	5
8:45	1152	8	768	5
8:50	1164	9	840	6
8:55	1128	7	828	5
9:00	1056	6	576	4
9:05	888	6	564	3
9:10	852	6	480	3
9:15	960	6	492	3
9:20	864	6	408	3
9:25	948	7	492	3
9:30	900	6	480	3
9:35	960	7	456	3
9:40	732	5	408	3
9:45	924	6	420	3
9:50	960	6	492	3
9:55	948	6	468	3
10:00	1008	7	468	3

DETECTOR STATION 479

detector 2634		detector 2635		detector 2636	
VOL	OCC	VOL	OCC	VOL	OCC
1080	8	984	6	648	4
972	7	1020	7	588	3
1164	8	1260	8	1044	6
1056	8	1488	10	1104	6
1380	10	1404	9	1104	6
1236	9	1368	9	1044	6
1392	10	1500	10	1320	8
1416	10	1512	10	1428	9
1320	10	1512	10	1308	8
1536	11	1500	10	1248	7
1380	10	1524	10	1200	7
1344	10	1668	11	1452	9
1464	11	1596	11	1452	8
1284	10	1272	9	1104	6
1332	10	1380	9	1212	7
1452	11	1488	10	1200	7
1296	9	1356	9	912	5
1452	11	1488	10	1128	7
1272	9	1476	10	1260	7
1140	8	1236	8	864	4
1284	9	1152	7	816	5
1260	10	1284	9	912	5
1260	10	1464	10	972	6
1176	9	1380	9	1032	6
1200	8	1212	8	756	4
1104	8	1188	8	792	4
1128	9	1068	7	504	3
1308	10	1068	8	804	5
1104	8	900	6	612	3
1212	10	996	7	624	3
1308	10	1104	7	600	3
1344	10	1080	7	540	3
1080	8	924	6	480	2
1068	8	948	6	600	3
1212	9	1092	7	612	3
1236	9	1044	8	612	3
1188	9	1152	8	540	3
1056	8	1032	7	528	3



DETECTOR STATION 489

TIME	detector 2638		detector 2639		detector 2640		detector 2641	
	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
6:55	660	3	864	4	864	4	708	2
7:00	516	2	828	4	792	3	540	2
7:05	780	4	864	4	912	4	828	3
7:10	576	2	948	4	1188	5	1032	4
7:15	816	4	1080	5	1296	6	1032	4
7:20	708	3	1188	6	1176	5	996	3
7:25	804	4	1272	6	1128	5	1188	4
7:30	1008	5	1164	6	1308	6	1248	5
7:35	924	4	1176	6	1224	6	1248	5
7:40	972	5	1176	5	1164	5	1080	4
7:45	1044	5	1452	7	1380	6	1080	4
7:50	960	4	1392	7	1308	6	1104	4
7:55	1116	5	1284	6	1440	6	1296	4
8:00	984	5	1188	6	1080	5	1116	4
8:05	900	4	1176	5	1176	5	948	3
8:10	900	4	1332	7	1248	5	1104	4
8:15	888	4	1080	5	1044	5	696	2
8:20	1044	5	1284	6	1260	6	1008	3
8:25	1056	5	1164	5	1140	5	912	3
8:30	852	4	1128	5	1056	4	828	3
8:35	768	3	1008	5	1140	5	768	2
8:40	912	4	1092	6	1128	5	696	2
8:45	696	3	1128	5	1128	5	852	3
8:50	816	4	1128	5	1164	5	792	2
8:55	852	4	1104	5	1116	5	672	2
9:00	756	3	900	4	936	4	648	2
9:05	708	3	912	4	972	4	408	1
9:10	696	3	1128	6	888	4	600	2
9:15	684	3	960	4	792	4	348	1
9:20	840	4	984	5	840	4	432	1
9:25	648	3	972	5	1068	4	516	2
9:30	672	3	1104	5	840	3	516	2
9:35	540	2	912	4	732	3	468	1
9:40	552	2	888	4	912	4	408	1
9:45	684	3	1140	5	804	3	540	2
9:50	684	3	1032	5	960	4	612	2
9:55	672	3	984	5	948	4	576	2
10:00	600	3	936	5	852	4	444	1

DETECTOR STATION 490

TIME	detector 2643		detector 2644		detector 2645		detector 2646	
	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
6:55	1068	7	924	7	996	6	660	4
7:00	1044	8	840	6	648	4	480	3
7:05	1128	8	792	5	912	6	720	4
7:10	1128	8	912	6	996	6	948	6
7:15	1524	11	984	7	1116	7	1164	7
7:20	1428	10	972	7	1116	7	996	6
7:25	1428	10	972	7	1116	7	996	6
7:30	1392	10	1008	8	1044	6	876	5
7:35	1752	12	1200	9	1104	8	1200	8
7:40	1560	11	1200	8	1188	7	1128	7
7:45	1812	14	1260	9	1212	7	1116	7
7:50	1800	14	1224	10	1248	8	1080	7
7:55	1860	14	1356	10	1332	8	1164	7
8:00	1860	14	1356	10	1332	8	1164	7
8:05	1632	11	1164	7	1164	8	972	6
8:10	1668	12	1212	9	1068	7	1104	7
8:15	1428	10	1116	8	1020	6	828	5
8:20	1824	13	1092	8	912	6	876	5
8:25	1740	12	1092	8	1200	8	888	5
8:30	1404	9	996	7	756	5	780	4
8:35	1200	8	900	6	972	6	804	5
8:40	1368	9	996	8	960	6	648	4
8:45	1308	8	1080	8	960	6	792	5
8:50	1440	11	1092	8	1008	8	816	5
8:55	1404	10	1068	8	1068	7	732	4
9:00	1044	7	876	6	912	6	660	4
9:05	1092	8	768	6	708	4	588	3
9:10	1320	10	852	7	708	5	552	3
9:15	1128	8	828	6	768	5	444	3
9:20	1068	8	708	6	756	5	480	3
9:25	1152	8	876	6	924	6	456	3
9:30	1200	8	864	6	672	5	552	4
9:35	960	6	780	6	636	4	588	3
9:40	960	6	888	6	672	4	576	3
9:45	1188	8	768	6	840	5	540	3
9:50	1080	8	816	6	852	5	660	4
9:55	1152	8	684	5	840	6	612	4
10:00	1044	7	948	7	756	5	564	3

DETECTOR STATION 491

TIME	detector 2648		detector 2649		detector 2650	
	VOL	OCC	VOL	OCC	VOL	OCC
6:55	864	7	876	6	816	5
7:00	756	6	636	5	600	4
7:05	828	6	744	5	660	4
7:10	684	5	876	6	792	5
7:15	1032	9	1176	8	1212	8
7:20	996	8	1080	8	900	6
7:25	960	8	1044	7	924	6
7:30	1188	10	1152	8	1092	7
7:35	960	7	1188	9	1140	9
7:40	1080	8	1056	7	1044	7
7:45	1224	9	1152	8	1152	8
7:50	1056	9	1176	8	1092	8
7:55	864	7	1188	8	1152	8
8:00	1224	11	1128	8	984	7
8:05	936	7	996	8	1092	8
8:10	936	8	1092	8	1008	7
8:15	972	7	840	6	864	6
8:20	876	7	936	6	792	6
8:25	936	8	996	7	1032	7
8:30	744	5	768	5	780	5
8:35	792	6	828	6	876	6
8:40	864	8	792	6	768	5
8:45	780	6	900	6	864	6
8:50	948	8	792	7	804	6
8:55	840	7	900	6	720	5
9:00	624	5	888	7	672	5
9:05	744	6	732	5	672	4
9:10	756	8	552	4	612	4
9:15	840	7	768	6	564	4
9:20	564	5	708	5	468	3
9:25	816	7	876	7	432	3
9:30	720	6	660	5	480	3
9:35	732	7	624	5	684	5
9:40	804	7	756	5	624	4
9:45	744	6	828	5	564	4
9:50	792	6	936	7	660	4
9:55	720	6	804	6	720	5
10:00	924	8	696	5	564	4

TIME	detector 722		detector 723		detector 434		detector 435		detector 95		detector 2191	
	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
6:55	0	100	252	3	456	3	828	7	1608	19	1644	6
7:00	0	100	180	2	216	1	756	6	2028	23	1596	5
7:05	0	100	264	2	216	1	756	6	1416	15	1752	6
7:10	0	100	264	3	300	2	840	7	1440	15	1788	6
7:15	0	100	396	4	456	3	984	9	1584	17	1752	6
7:20	0	100	240	2	396	3	1272	12	1668	20	1908	7
7:25	0	100	288	3	504	4	1092	10	1956	25	1728	6
7:30	0	100	300	3	444	3	1308	12	1956	23	1908	7
7:35	0	100	408	4	552	5	1392	13	1836	20	1896	6
7:40	0	100	336	3	552	5	1392	13	1812	20	2028	7
7:45	0	100	372	4	504	4	1608	15	1872	21	1860	6
7:50	0	100	456	4	504	4	1344	14	1884	21	1956	7
7:55	0	100	324	3	396	3	1200	11	2040	26	1980	7
8:00	0	100	240	2	396	3	1200	11	1740	19	2076	7
8:05	0	100	276	3	408	3	1104	10	1872	20	1944	7
8:10	0	100	228	2	480	4	972	10	1680	18	2088	7
8:15	0	100	204	2	492	4	1188	10	1536	16	1728	6
8:20	0	100	252	2	492	4	1188	10	1896	21	1908	6
8:25	0	100	180	1	432	3	1032	8	1668	17	2016	7
8:30	0	100	228	2	384	2	1068	8	1728	18	1740	6
8:35	0	100	216	2	384	2	1068	8	1788	20	1656	6
8:40	0	100	252	2	384	3	1044	9	1692	19	1704	6
8:45	0	100	252	2	408	3	960	7	1368	14	1860	6
8:50	0	100	240	2	528	3	1308	11	1572	17	1752	6
8:55	0	100	216	2	480	3	960	8	1632	18	1860	6
9:00	0	100	216	2	468	3	1140	9	1536	16	1692	6
9:05	0	100	192	2	372	2	924	7	1428	16	1644	6
9:10	0	100	132	1	360	2	744	6	1104	11	1272	4
9:15	0	100	276	2	456	3	852	7	1164	12	1308	4
9:20	0	100	216	2	540	4	744	6	1068	13	1416	5
9:25	0	100	216	2	456	3	780	6	1140	12	1464	5
9:30	0	100	180	1	288	2	660	6	1128	13	1416	5
9:35	0	100	204	2	372	2	696	5	1116	13	1500	5
9:40	0	100	96	1	276	1	780	6	1188	13	1416	5
9:45	0	99	120	1	384	3	768	6	1080	12	1560	5
9:50	0	100	108	1	468	3	900	7	1536	18	1548	5
9:55	0	100	96	1	384	3	900	8	1344	16	1524	5
10:00	0	100	84	0	324	2	696	5	1284	15	1464	5

TIME	detector 2604		detector 2605		detector 2609		detector 2610		detector 2615		detector 2616	
	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
6:55	156	1	0	0	624	4	204	3	156	3	0	0
7:00	120	1	0	0	708	7	324	5	132	2	0	0
7:05	108	1	0	0	708	4	168	2	156	3	0	0
7:10	216	2	0	0	648	5	324	5	204	4	0	0
7:15	156	1	0	0	744	5	264	4	192	4	0	0
7:20	228	3	0	0	696	6	360	5	228	4	0	0
7:25	168	1	0	0	720	5	300	5	228	5	0	0
7:30	216	2	0	0	684	4	372	6	228	4	0	0
7:35	144	1	0	0	600	4	480	8	156	3	0	0
7:40	300	3	0	0	588	4	396	7	288	6	0	0
7:45	276	4	0	0	780	5	348	5	204	3	0	0
7:50	216	3	0	0	636	4	456	7	240	5	0	0
7:55	216	2	0	0	720	4	312	4	216	4	0	0
8:00	252	2	0	0	636	4	408	6	252	5	0	0
8:05	180	2	0	0	612	4	504	8	216	5	0	0
8:10	168	2	0	0	720	5	372	6	264	6	0	0
8:15	204	2	0	0	708	5	384	7	168	3	0	0
8:20	168	2	0	0	756	5	468	8	252	6	0	0
8:25	168	1	0	0	696	5	516	9	204	4	0	0
8:30	216	2	0	0	768	5	264	4	276	7	0	0
8:35	252	2	0	0	660	5	408	6	252	5	0	0
8:40	228	2	0	0	768	5	336	5	276	6	0	0
8:45	228	2	0	0	768	5	264	4	312	7	0	0
8:50	120	1	0	0	684	5	360	5	264	5	0	0
8:55	204	1	0	0	756	6	528	9	216	4	0	0
9:00	192	1	0	0	744	5	456	7	240	5	0	0
9:05	204	1	0	0	804	5	396	6	228	5	0	0
9:10	192	1	0	0	792	6	468	7	204	5	0	0
9:15	192	1	0	0	780	5	432	7	252	7	0	0
9:20	120	1	0	0	804	5	372	6	360	9	0	0
9:25	120	1	0	0	984	7	468	9	348	8	0	0
9:30	60	0	0	0	708	6	312	6	300	7	0	0
9:35	240	2	0	0	948	7	480	8	312	7	0	0
9:40	264	2	0	0	900	7	528	9	264	8	0	0
9:45	204	2	0	0	1068	9	540	10	336	9	0	0
9:50	156	1	0	0	936	7	432	7	324	10	0	0
9:55	144	1	0	0	816	6	516	7	324	12	0	0
10:00	144	1	0	0	660	5	528	9	348	8	0	0

TIME	detector 2624		detector 2628		detector 2629		detector 2637	
	VOL	OCC	VOL	OCC	VOL	OCC	VOL	OCC
6:55	60	0	396	3	396	4	588	5
7:00	84	0	384	4	348	3	288	2
7:05	120	1	540	5	348	3	624	5
7:10	204	3	756	8	552	6	648	6
7:15	144	2	540	7	636	8	600	5
7:20	156	1	564	5	648	8	576	5
7:25	48	0	552	5	588	6	636	5
7:30	180	2	648	6	564	6	756	7
7:35	144	1	612	6	684	8	612	5
7:40	120	1	684	7	672	8	936	8
7:45	240	3	708	6	648	7	936	8
7:50	240	3	636	6	648	7	1008	9
7:55	312	4	684	7	816	9	1008	9
8:00	120	1	744	7	876	12	1020	9
8:05	204	3	768	7	744	8	852	7
8:10	180	2	780	7	948	12	816	7
8:15	144	2	648	6	756	8	756	6
8:20	180	3	624	6	624	6	840	7
8:25	132	2	672	7	804	10	852	7
8:30	156	1	600	6	432	5	744	6
8:35	168	1	672	7	660	8	804	7
8:40	216	2	852	8	552	8	744	6
8:45	120	1	840	8	564	6	504	5
8:50	216	2	504	5	564	7	852	8
8:55	324	3	792	7	684	9	792	7
9:00	216	2	624	6	516	5	660	6
9:05	312	3	672	6	480	6	756	7
9:10	156	1	600	7	288	3	576	4
9:15	180	2	660	7	456	6	468	4
9:20	192	2	720	8	348	4	648	5
9:25	216	2	684	7	408	6	492	4
9:30	180	1	780	8	408	5	600	6
9:35	228	2	660	7	324	3	468	4
9:40	204	2	744	9	468	6	600	5
9:45	204	2	636	7	468	5	684	6
9:50	144	1	780	10	384	4	720	7
9:55	240	2	828	9	468	6	648	6
10:00	312	3	612	7	360	4	492	4

TIME	detector2642		detector 2647		detector 2651	
	VOL	OCC	VOL	OCC	VOL	OCC
6:55	504	5	336	2	360	4
7:00	408	4	228	1	336	3
7:05	336	3	288	2	348	3
7:10	360	3	216	1	300	3
7:15	624	5	384	3	432	4
7:20	540	4	408	4	432	4
7:25	624	6	408	4	384	4
7:30	336	3	492	4	504	5
7:35	720	6	288	2	588	6
7:40	756	7	504	4	600	6
7:45	696	6	444	3	648	6
7:50	780	7	492	4	552	5
7:55	840	7	444	3	732	7
8:00	768	7	444	3	744	7
8:05	612	5	348	3	624	6
8:10	816	8	384	3	564	5
8:15	672	6	324	2	420	4
8:20	600	5	408	3	768	8
8:25	408	3	468	4	600	6
8:30	648	6	288	2	708	7
8:35	588	5	324	3	492	5
8:40	528	4	312	2	456	5
8:45	672	6	348	2	420	4
8:50	576	5	420	4	504	5
8:55	468	4	432	4	756	8
9:00	468	4	276	2	336	3
9:05	444	4	264	2	336	3
9:10	372	3	336	3	516	5
9:15	552	6	264	2	360	3
9:20	444	4	204	2	336	3
9:25	432	4	300	2	396	4
9:30	456	4	240	2	264	3
9:35	480	4	288	2	252	2
9:40	516	5	516	4	288	3
9:45	540	5	372	3	192	2
9:50	408	3	336	2	288	2
9:55	384	3	300	2	180	1
10:00	492	4	336	3	252	2

## Calibrations for I-94 EB During the University of Minnesota Ramp Construction

### FRESIM :

- default vehicle set.
- car following factors = 15 to 6.
- acceleration lag = 0.3 sec.
- deceleration lag = 0.3 sec.
- lane change time = 3 sec.
- vehicle generation minimum separation = 1.0 sec.
- percentage of drivers willing to yield = 20 percent.
- minimum non-emergency deceleration = 0.8 ft/sec<sup>2</sup>

### A IMSUN2 :

#### vehicle characteristics:

- length = 4.8 m.
- width = 1.8 m.
- desired speed = 104 kph.
- maximum acceleration = 2.8 m/sec<sup>2</sup>
- maximum deceleration = 3.5 m/sec<sup>2</sup>

#### simulation parameters:

- simulation step size = 0.75sec
- speed acceptance = 0.80
- braking coefficient = 2.0
- zone 1 distance = 300 m
- zone 2 distance = 75 m
- minimum distance between vehicles = 1.2 m
- queue speed = 1.0 m/sec<sup>2</sup>
- queue leaving speed = 4.0 m/sec<sup>2</sup>
- junction visibility = 25 m
- cruising tolerance = 0.80

### FREFLO :

- relaxation time = 75 sec/mile
- anticipation coefficient = 0.25 miles<sup>2</sup>/hour
- capacities as listed on the freeway geometrics
- cubic polynomial speed-density relationship

$$\text{speed} = 79 - 102 (k * 10^{-2}) + 68 (k * 10^{-2})^2 - 26 (k * 10^{-2})^3$$



FREQ11:

- capacities as listed on the freeway geometrics
- points on the speed-V/C curve

Points on Upper Portion		of Speed-V/C curve	Points on Lower Portion		of Speed-V/C curve
V/C ratio	speed (mph)		V/C ratio	speed (mph)	
0.00	65		0.00	0	
0.44	65		0.25	8	
0.71	57		0.65	21	
0.81	48		0.94	31	
0.91	41		1.00	34	
0.96	36				
1.00	34				

KRONOS:

- jam density = 130 vpm
- maximum capacity = 2100 vph
- density at maximum capacity = 45 vpm
- density at end of linear region of QK curve = 18 vpm
- flow rate at end of linear region of QK curve = 1175 vph
- capacities as listed on the freeway geometrics

INTEGRATION:

- mainline freeflow speed = 104 kph
- ramp freeflow speed for capacities less than 2100 vph = 72 kph
- ramp freeflow speed when capacity is 2100 vph = 104 kph
- capacities as listed on the freeway geometrics
- speed at capacity for mainline and 2100 vph capacity ramps = 72 kph
- speed at capacity for all other ramps = 56.4 kph
- mainline jam density when capacity is 2400 vphpl = 91.7 vpkpl
- mainline jam density when capacity is 2100 vphpl = 80.3 vpkpl
- ramp jam density when capacity is 2100 vph = 80.3 vpkpl
- all other ramp jam densities = 130 vpkpl

## Appendix B

### Qualitative Test Cases

Geometrics for I-494 Westbound at the TH-100 Cloverleaf	B-3
Flow Data for Test Case 1-A : I-494 Congestion Spillback	B-4
Calibration Parameters for Test Case 1-A : I-494 Congestion Spillback	B-5
Flow Data for Test Case 1-B : I-494 Conservation	B-7
Calibration Parameters for Test Case 1-B : I-494 Congestion	B-8
Density Tables for Test Case 1-A : I-494WB Congestion Spillback	B-10
Geometrics for I-35W Through Burnsville	B-16
Flow Data for Test Case 2-A : I-35W Congestion Spillback	B-17
Calibration Parameters for Test Case 2-A : I-35W Congestion Spillback	B-18
Flow Data for Test Case 2-B : I-35W Incident Spillback	B-20
Calibration Parameters for Test Case 2-B : I-35W Incident Spillback	B-21
Density Tables for Test Case 2-A : I-35W Congestion Spillback	B-23
Density Tables for Test Case 2-B : I-35W Incident Spillback	B-30

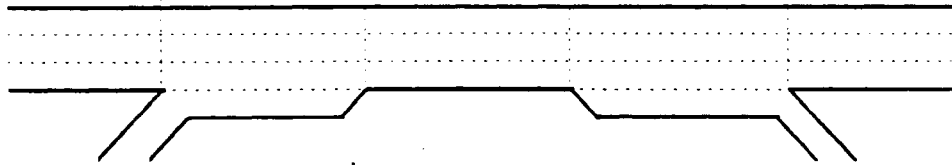
## Appendix B

### Qualitative Test Cases

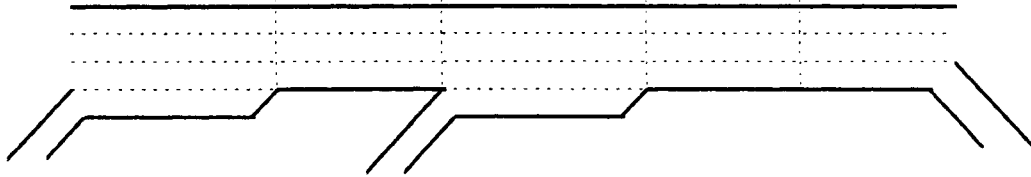
Geometrics for Multiple Lane Drop Freeway	B-38
Flow Data for Test Case 3-A : Multiple Lane Drop Congestion Spillback	B-39
Calibration Parameters for Test Case 3-A : Multiple Lane Drop Congestion Spillback	B-40
Flow Data for Test Case 3-B : Multiple Lane Drop Incident Spillback	B-42
Calibration Parameters for Test Case 3-B : Multiple Lane Drop Incident Spillback	B-43
Density Tables for Test Case 3-A : Multiple Lane Drop Congestion Spillback	B-45
Density Tables for Test Case 3-B : Multiple Lane Drop Incident Spillback	B-52

## I-494 Westbound at the TH-100 Cloverleaf

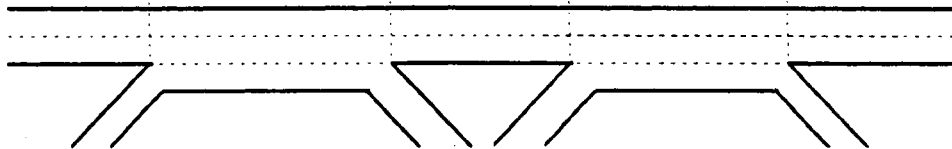
cap : 6300 vph len : 1390 ft 0.421 km	cap : 7800 vph len : 600 ft 0.182 km	cap : 6300 vph len : 2340 ft 0.709 km	cap : 7800 vph len : 250 ft 0.076 km	cap : 6300 vph len : 1550 ft 0.470 km
	ramp cap : 1500 vph ramp len : 800 ft 0.242 km		ramp cap : 1500 vph ramp len : 1500 ft 0.455 km	



cap : 7800 vph len : 400 ft 0.121 km	cap : 6300 vph len : 750 ft 0.227 km	cap : 7800 vph len : 500 ft 0.152 km	cap : 6300 vph len : 1925 ft 0.583 km	cap : 7800 vph len : 1185 ft 0.359 km
ramp cap : 1500 vph ramp len : 1500 ft 0.455 km		ramp cap : 1500 vph ramp len : 600 ft 0.182 km		ramp cap : 1500 vph ramp len : 500 ft 0.152 km



cap : 4800 vph len : 800 ft 0.242 km	cap : 6300 vph len : 400 ft, 0.121 km	cap : 4800 vph len : 810 ft 0.245 km	cap : 6300 vph len : 1015 ft, 0.308 km	cap : 4400 vph len : 2805 ft 0.850 km
	ramp cap : 1500 vph on ramp len : 600 ft 0.182 km off ramp len : 500 ft 0.152 km		ramp cap : 1500 vph on ramp len : 600 ft 0.182 km off ramp len : 500 ft 0.152 km	



Flow Data for Test Case 1-A : I-494 Congestion Spillback

time slice	mainline vol. (vph)	ramp 1 vol. (vph)	ramp 2 vol. (vph)	ramp 3 vol. (vph)	ramp 4 vol. (vph)
1	4500	300	300	300	300
2	4500	300	300	300	300
3	4500	300	300	300	300
4	4500	300	300	300	300
5	4500	300	300	300	300
6	4500	300	300	300	300
7	4500	300	300	300	300
8	4500	300	300	300	300
9	4500	300	300	300	300
10	4500	300	300	300	300
11	2000	300	300	300	300
12	2000	300	300	300	300
13	2000	300	300	300	300
14	2000	300	300	300	300
15	2000	300	300	300	300
16	2000	300	300	300	300
17	2000	300	300	300	300
18	2000	300	300	300	300

time slice	ramp 5 vol. (vph)	ramp 6 vol. (vph)	ramp 7 vol. (vph)	ramp 8 vol. (vph)	ramp 9 vol. (vph)
1	600	300	300	300	1200
2	600	300	300	300	1200
3	600	1200	1200	300	1200
4	600	1200	1200	300	1200
5	600	1200	1200	300	1200
6	600	1200	1200	300	1200
7	600	1200	1200	300	1200
8	600	1200	1200	300	1200
9	600	1200	1200	300	1200
10	600	1200	1200	300	1200
11	600	300	300	300	1200
12	600	300	300	300	1200
13	600	300	300	300	1200
14	600	300	300	300	1200
15	600	300	300	300	1200
16	600	300	300	300	1200
17	600	300	300	300	1200
18	600	300	300	300	1200

## Calibration Parameters for Test Case 1-A : I-494 Congestion Spillback

### FRESIM :

- default vehicle set.
- car following factors = 15 to 6.
- acceleration lag = 0.3 sec.
- deceleration lag = 0.3 sec.
- lane change time = 3 sec.
- vehicle generation minimum separation = 1.0 sec.
- percentage of drivers willing to yield = 30 percent.
- minimum non-emergency deceleration = 0.8 ft/sec<sup>2</sup>

### AIMSUN2 :

#### vehicle characteristics:

- length = 4.0 m.
- width = 1.7 m.
- desired speed = 104 kph.
- maximum acceleration = 2.8 m/sec<sup>2</sup>
- maximum deceleration = 4.0 m/sec<sup>2</sup>

#### simulation parameters:

- simulation step size = 0.75sec
- speed acceptance = 0.80
- braking coefficient = 2.0
- zone 1 distance = 120 m
- zone 2 distance = 40 m
- minimum distance between vehicles = 1.2 m
- queue speed = 1.0 m/sec<sup>2</sup>
- queue leaving speed = 4.0 m/sec<sup>2</sup>
- junction visibility = 25 m
- cruising tolerance = 0.80

### FREFLO :

- relaxation time = 75 sec/mile
- anticipation coefficient = 0.25 miles<sup>2</sup>/hour
- capacities as listed on the freeway geometrics
- cubic polynomial speed-density relationship

$$\text{speed} = 87 - 49 (k * 10^{-2}) - 100 (k * 10^{-2})^2 + 64 (k * 10^{-2})^3$$

**FREQ11:**

- capacities as listed on the freeway geometrics
- points on the speed-V/C curve

Points on Upper Portion		of Speed-V/C curve	Points on Lower Portion		of Speed-V/C curve
V/C ratio	speed (mph)		V/C ratio	speed (mph)	
0.00	65		0.00	0	
0.71	65		0.34	9	
0.92	44		0.78	21	
1.00	34		1.00	34	

**KRONOS:**

- jam density = 136 vpm
- maximum capacity = 2200 vph
- density at maximum capacity = 41 vpm
- density at end of linear region of QK curve = 24 vpm
- flow rate at end of linear region of QK curve = 1580 vph
- capacities as listed on the freeway geometrics

**INTEGRATION:**

- mainline freeflow speed = 104 kph
- ramp freeflow speed for capacities less than 2100 vph = 89 kph
- ramp freeflow speed when capacity is 2100 vph = 104 kph
- capacities as listed on the freeway geometrics
- speed at capacity for mainline and 2100 vph capacity ramps = 72 kph
- speed at capacity for all other ramps = 56.4 kph
- mainline jam density when capacity is 2400 vphpl = 91.7 vpkpl
- mainline jam density when capacity is 2300 vphpl = 87.9 vpkpl
- mainline jam density when capacity is 2200 vpkpl = 85.1 vpkpl
- ramp jam density when capacity is 2100 vph = 80.3 vpkpl
- all other ramp jam densities = 130 vpkpl

Flow Data for Test Case 1- B : I-494 Conservation

time slice	mainline vol. (vph)	ramp 1 vol. (vph)	ramp 2 vol. (vph)	ramp 3 vol. (vph)	ramp 4 vol. (vph)
1	2200	300	300	300	300
2	2200	300	300	300	300
3	2200	300	300	300	300
4	2200	600	600	300	300
5	2200	600	600	300	300
6	2200	600	600	300	300
7	4400	600	600	600	600
8	4400	600	600	600	600
9	4400	600	600	600	600
10	4400	300	300	600	600
11	4400	300	300	600	600
12	4400	300	300	600	600
13	1360	300	300	300	300
14	1360	300	300	300	300
15	1360	300	300	300	300
16	1360	600	600	300	300
17	1360	600	600	300	300
18	1360	600	600	300	300

time slice	ramp 5 vol. (vph)	ramp 6 vol. (vph)	ramp 7 vol. (vph)	ramp 8 vol. (vph)	ramp 9 vol. (vph)
1	600	600	300	300	600
2	600	600	300	300	600
3	600	600	300	300	600
4	600	600	300	300	600
5	600	600	300	300	600
6	600	600	300	300	600
7	1200	300	300	300	300
8	1200	300	300	300	300
9	1200	300	300	300	300
10	1200	300	300	300	300
11	1200	300	300	300	300
12	1200	300	300	300	300
13	600	300	600	600	300
14	600	300	600	600	300
15	600	300	600	600	300
16	600	300	600	600	300
17	600	300	600	600	300
18	600	300	600	600	300



## Calibration Parameters for Test Case 1-B : I-494 Conservation

### FRESIM :

- default vehicle set.
- car following factors = 15 to 6.
- acceleration lag = 0.3 sec.
- deceleration lag = 0.3 sec.
- lane change time = 3 sec.
- vehicle generation minimum separation = 1.0 sec.
- percentage of drivers willing to yield = 30 percent.
- minimum non-emergency deceleration = 0.8 ft/sec<sup>2</sup>

### AIMSUN2 :

#### vehicle characteristics:

- length = 4.0 m.
- width = 1.7 m.
- desired speed = 104 kph.
- maximum acceleration = 2.8 m/sec<sup>2</sup>
- maximum deceleration = 4.0 m/sec<sup>2</sup>

#### simulation parameters:

- simulation step size = 0.75sec
- speed acceptance = 0.80
- braking coefficient = 2.0
- zone 1 distance = 120 m
- zone 2 distance = 40 m
- minimum distance between vehicles = 1.2 m
- queue speed = 1.0 m/sec<sup>2</sup>
- queue leaving speed = 4.0 m/sec<sup>2</sup>
- junction visibility = 25 m
- cruising tolerance = 0.80

### FREFLO :

- relaxation time = 75 sec/mile
- anticipation coefficient = 0.25 miles<sup>2</sup>/hour
- capacities as listed on the freeway geometrics
- cubic polynomial speed-density relationship

$$\text{speed} = 87 - 49 (k * 10^{-2}) - 100 (k * 10^{-2})^2 + 64 (k * 10^{-2})^3$$

**FREQ11:**

- capacities as listed on the freeway geometrics
- points on the speed-V/C curve

Points on Upper Portion of Speed-V/C curve		Points on Lower Portion of Speed-V/C curve	
V/C ratio	speed (mph)	V/C ratio	speed (mph)
0.00	65	0.00	0
0.71	65	0.34	9
0.92	44	0.78	21
1.00	34	1.00	34

**KRONOS:**

- jam density = 136 vpm
- maximum capacity = 2200 vph
- density at maximum capacity = 41 vpm
- density at end of linear region of QK curve = 24 vpm
- flow rate at end of linear region of QK curve = 1580 vph
- capacities as listed on the freeway geometrics

**INTEGRATION:**

- mainline freeflow speed = 104 kph
- ramp freeflow speed for capacities less than 2100 vph = 89 kph
- ramp freeflow speed when capacity is 2100 vph = 104 kph
- capacities as listed on the freeway geometrics
- speed at capacity for mainline and 2100 vph capacity ramps = 72 kph
- speed at capacity for all other ramps = 56.4 kph
- mainline jam density when capacity is 2400 vphpl = 91.7 vpkpl
- mainline jam density when capacity is 2300 vphpl = 87.9 vpkpl
- mainline jam density when capacity is 2200 vpkpl = 85.1 vpkpl
- ramp jam density when capacity is 2100 vph = 80.3 vpkpl
- all other ramp jam densities = 130 vpkpl

Density Tables for Test Case 1-A : I-494 WB Congestion Spillback

FRESIM																		
hypothetical test case																		
1-A																		
DENSITY in vehicles/mile/lane																		
DIST	TIMESLICE																	
ft	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
5180	25	25	25	25	24	25	24	25	24	25	18	11	11	11	11	11	11	11
5380	25	25	25	25	25	25	25	25	25	25	18	11	11	11	11	10	11	11
5580	25	25	25	26	25	25	24	26	25	25	19	11	11	11	11	10	11	11
5780	25	25	25	26	25	25	25	26	25	25	19	11	11	11	11	10	11	11
5980	26	25	24	26	25	25	24	26	25	25	19	10	11	11	11	10	11	11
6130	26	25	25	26	25	25	25	26	25	25	19	11	11	11	11	11	11	11
6330	28	27	27	28	27	27	27	28	27	28	21	13	13	13	13	12	13	13
6430	21	21	20	21	20	21	20	21	20	21	16	10	10	9	10	9	10	9
6680	22	21	20	21	21	20	21	21	20	21	16	9	9	9	9	9	9	9
6880	28	28	27	27	28	27	27	28	27	28	22	13	12	12	12	12	13	12
7080	28	28	27	27	28	27	27	27	27	27	23	12	12	12	12	12	13	12
7280	28	27	27	27	28	27	27	27	27	27	23	12	12	12	12	12	13	12
7480	30	29	29	29	29	29	29	29	29	29	25	14	14	14	14	14	14	14
7680	22	22	22	22	22	21	22	22	21	22	19	11	11	10	10	10	11	10
7905	22	21	21	22	21	21	22	22	21	21	19	10	10	10	10	10	10	10
8105	30	28	28	28	28	27	28	28	28	28	25	14	14	14	13	14	13	14
8305	29	27	28	28	28	27	28	28	27	28	25	14	14	14	13	14	13	14
8505	29	27	28	27	28	27	28	28	27	27	26	14	14	13	13	13	14	14
8705	29	27	27	27	28	27	28	28	27	27	26	13	14	14	13	14	14	13
8905	29	27	27	27	28	27	28	27	27	27	26	13	14	13	13	14	13	13
9105	29	27	27	27	28	27	28	28	27	27	26	13	14	13	13	13	14	13
9305	29	28	27	27	28	27	28	28	27	28	27	14	14	13	13	14	14	13
9505	29	28	27	28	28	28	28	28	27	27	27	14	14	13	13	14	13	14
9705	30	28	28	28	29	28	28	28	28	28	28	14	14	14	13	14	13	14
9890	31	29	29	29	29	29	29	29	29	29	29	14	14	14	13	14	13	14
10090	32	30	30	30	30	29	30	30	30	29	30	14	14	13	14	14	14	14
10290	32	31	31	31	31	30	30	29	30	30	30	14	14	14	14	14	13	14
10490	33	31	31	32	31	31	31	30	30	31	31	14	14	14	13	14	13	14
10690	35	33	32	35	32	32	32	31	31	32	32	15	14	14	13	14	14	14
10890	54	50	50	54	51	49	49	48	48	49	49	23	21	21	20	21	21	21
11090	45	43	44	47	43	42	44	42	42	43	42	18	17	15	16	16	16	16
11290	44	43	44	46	42	40	42	42	42	41	41	19	17	15	16	16	16	16
11490	43	44	43	47	41	39	41	40	41	41	39	19	17	15	16	16	15	16
11690	43	45	43	49	41	39	41	40	40	40	39	20	17	15	16	16	15	16
12090	47	50	58	67	55	52	54	53	52	54	44	23	20	18	19	19	18	19
12300	45	48	49	52	42	38	40	41	39	41	35	20	17	15	15	16	15	16
12500	45	46	46	48	41	37	40	40	38	40	35	21	16	16	15	16	15	15
12700	47	45	44	47	41	36	39	40	38	38	35	21	16	15	15	16	14	15
12900	48	44	43	47	39	37	38	39	37	38	35	21	16	15	15	16	14	15
13100	50	48	46	49	41	40	40	41	40	39	37	25	18	18	18	19	17	18
13300	33	32	31	32	27	27	26	27	26	26	24	16	12	12	12	12	11	12
13515	31	31	30	30	27	26	26	26	25	26	23	16	12	12	12	12	11	12
13715	31	29	29	29	26	26	25	25	25	25	23	16	11	12	12	12	11	11
13915	45	43	44	42	38	38	37	36	37	38	34	24	17	17	17	18	16	18
14115	41	41	41	39	35	35	34	34	35	36	30	22	14	15	14	15	14	14

AIMSUN2																		
hypothetical test case																		
1-A																		
DENSITY in vehicles/mile/lane																		
DIST ft	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
5180	11	10	14	12	14	14	13	15	12	8	6	8	8	6	7	7	6	8
5380	11	10	13	12	14	14	13	15	11	8	6	8	9	6	6	8	6	7
5580	11	10	13	12	14	14	13	15	11	8	6	8	9	6	7	8	6	7
5780	11	11	13	12	14	14	13	15	12	8	6	7	9	6	6	7	6	7
5980	11	11	13	12	13	14	13	15	12	9	6	7	9	6	6	7	6	7
6130	11	11	12	12	14	14	14	14	12	9	6	7	9	6	7	8	6	7
6330	10	9	11	11	11	12	11	12	10	7	5	7	7	6	7	7	6	7
6530	11	9	12	11	13	13	12	12	12	8	6	7	8	6	8	7	6	7
6680	13	11	14	14	14	16	15	16	16	10	7	8	9	8	10	8	8	9
6880	13	11	14	14	15	16	16	15	15	10	7	8	10	7	9	8	7	8
7080	12	12	14	13	15	15	15	16	14	10	7	8	10	7	9	8	7	8
7280	12	12	14	13	15	15	15	15	14	10	7	8	10	7	8	8	7	8
7530	11	11	12	12	13	13	13	12	12	9	7	7	10	6	7	8	7	9
7780	11	12	12	13	14	14	15	13	15	9	7	8	11	6	7	8	8	9
7905	15	17	15	19	18	19	22	19	23	12	9	10	16	8	9	10	10	12
8105	14	14	15	16	17	16	18	17	18	11	9	9	13	8	9	9	9	10
8305	13	14	14	15	16	15	18	17	20	11	8	9	12	8	9	9	9	9
8505	13	13	14	15	15	15	17	17	27	14	9	9	12	8	9	9	9	9
8705	13	13	14	15	15	15	17	19	52	29	9	9	12	8	9	9	9	9
8905	13	13	14	15	16	14	19	23	79	64	11	9	11	8	9	8	9	9
9105	13	13	14	15	16	14	23	35	99	90	21	9	11	8	9	8	9	9
9305	12	13	14	15	16	14	32	75	103	97	45	9	11	8	9	8	8	10
9505	13	13	14	14	16	15	55	100	103	96	66	9	11	8	10	8	8	10
9705	12	13	14	15	19	19	80	103	102	97	79	9	11	8	10	8	9	9
9890	12	14	14	18	23	30	95	103	101	97	89	14	11	8	10	8	9	10
10090	13	17	16	22	37	71	106	103	102	95	91	27	11	8	10	7	9	10
10290	15	26	21	43	69	107	105	104	101	95	91	41	11	8	10	7	9	10
10490	24	56	45	76	94	112	104	104	99	95	90	61	11	9	10	8	9	10
10690	49	92	87	96	98	112	106	105	98	95	91	84	16	10	10	8	9	12
10890	39	59	60	62	59	78	69	67	62	61	60	57	25	14	13	9	11	18
11090	22	27	28	27	40	80	70	67	58	51	30	30	20	12	14	9	12	13
11290	18	22	22	22	52	89	77	76	79	64	34	33	18	10	14	9	11	11
11490	17	20	19	24	67	90	78	79	86	67	43	44	19	10	13	9	10	11
11690	17	20	20	36	85	90	79	80	84	65	53	56	24	10	13	9	11	11
11890	13	16	18	54	87	78	78	82	86	67	66	66	38	8	10	7	8	10
12090	14	17	25	48	58	53	57	57	62	55	58	53	47	11	11	8	9	10
12300	17	21	17	25	24	27	25	27	22	28	27	27	25	12	14	9	11	11
12500	17	20	16	20	20	21	20	22	19	23	22	22	21	12	13	9	10	10
12700	16	20	15	18	18	20	19	20	17	21	20	21	19	12	12	9	10	10
12900	17	21	15	18	18	19	18	19	17	21	19	21	19	12	11	9	10	10
13100	14	18	13	13	14	14	13	14	13	15	14	16	13	10	9	8	9	8
13300	17	24	16	13	13	13	13	13	12	14	14	15	13	9	9	7	8	8
13515	26	60	36	14	12	13	13	13	12	14	13	15	13	9	8	7	7	8
13715	43	101	73	19	13	13	13	13	12	14	14	15	13	9	8	7	7	8
13915	35	60	50	29	14	13	14	13	12	14	17	18	16	12	8	8	8	8
14115	20	27	28	24	19	18	18	18	16	17	20	21	16	14	10	11	10	11

FREFLO																	
hypothetical test case																	
1-A																	
DENSITY in vehicles/mile/lane																	
DIST ft.	TIMESLICE																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
5180	23	23	23	23	23	23	23	23	23	23	20	10	10	10	10	10	10
5380	23	23	23	23	23	23	23	23	23	23	21	10	10	10	10	10	10
5580	23	23	23	23	23	23	23	23	23	23	21	11	10	10	10	10	10
5780	23	23	23	23	23	23	23	23	23	23	21	11	10	10	10	10	10
5930	23	23	23	23	23	23	23	23	23	23	22	11	10	10	10	10	10
6130	19	19	19	19	19	19	19	19	19	19	21	10	9	9	9	9	9
6330	19	19	19	19	19	19	19	19	19	23	54	12	9	9	9	9	9
6480	25	25	25	25	25	25	25	25	25	39	86	22	12	12	12	12	12
6680	25	25	25	25	25	25	25	25	25	53	80	22	12	12	12	12	12
6880	25	25	25	25	25	25	25	25	26	76	73	26	12	12	12	12	12
7080	24	25	25	24	25	25	25	25	36	87	67	45	12	12	12	12	12
7280	20	20	20	20	20	20	20	24	52	76	63	51	10	10	10	10	10
7580	20	20	20	20	20	20	65	96	77	74	63	10	10	10	10	10	10
7705	26	27	27	26	27	27	28	108	106	86	98	83	14	13	14	13	14
7905	26	26	26	26	26	26	44	91	80	70	79	70	13	13	13	13	13
8105	26	26	26	26	26	26	81	76	68	63	68	62	13	13	13	13	13
8305	26	26	26	26	26	29	93	68	65	61	65	57	14	13	13	13	13
8505	26	26	26	26	26	51	86	64	64	65	68	47	14	13	13	13	13
8705	26	26	26	26	26	73	74	60	63	68	66	43	15	13	13	13	13
8905	26	26	26	26	28	87	67	59	62	73	61	39	16	13	13	13	13
9105	26	26	26	26	40	83	63	59	62	76	59	36	18	13	13	13	13
9305	26	26	26	26	65	75	62	58	67	73	58	35	34	13	13	13	13
9505	26	26	26	27	90	69	59	58	71	67	57	34	43	13	13	13	13
9690	26	26	26	34	87	62	61	60	76	65	60	34	47	13	13	13	13
9890	26	26	27	68	77	62	63	70	76	65	58	38	47	13	13	13	13
10090	26	26	34	84	70	61	62	74	73	63	58	43	40	13	13	13	13
10290	26	26	50	91	65	60	66	76	71	64	59	51	36	13	13	13	13
10490	26	27	64	81	67	65	77	76	70	68	62	55	33	13	13	13	13
10690	29	31	71	74	72	74	82	77	72	74	66	63	35	13	13	13	13
10890	63	82	95	82	89	94	93	89	86	93	75	94	57	15	15	15	15
11090	54	73	73	58	62	65	64	61	61	64	57	71	51	15	15	15	15
11290	47	56	51	46	48	49	49	47	47	49	47	51	45	15	15	15	15
11490	41	44	41	40	40	41	41	40	40	41	40	41	39	15	15	15	15
11890	28	28	31	32	32	32	32	32	32	32	27	27	26	13	12	12	12
12100	38	39	35	36	36	37	37	36	36	37	35	33	33	17	16	15	16
12300	38	38	34	36	35	35	36	36	35	36	34	32	32	18	15	15	15
12500	37	37	33	35	35	35	35	35	34	35	34	31	31	18	15	15	15
12700	35	35	32	33	33	33	33	33	33	33	32	30	30	18	15	15	15
12900	25	25	23	24	24	24	24	23	24	24	23	22	22	15	12	12	12
13100	24	24	23	23	23	23	23	23	23	23	23	22	22	15	12	12	12
13315	24	24	23	23	23	23	23	23	23	23	23	22	22	15	12	12	12
13515	24	24	23	23	23	23	23	23	23	23	23	22	22	15	12	12	12
13715	25	25	23	24	24	24	24	24	24	24	23	22	22	16	12	12	12
13915	35	35	33	33	33	33	33	33	33	33	30	28	28	21	15	15	15
14115	36	36	33	33	34	33	34	34	34	34	30	29	28	21	15	15	15

FREQ11

hypothetical test case  
1-A

DENSITY in vehicles/mile/lane

DIST ft	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
600	23	23	23	23	23	23	23	23	23	23	23	10	10	10	10	10	10	10
800	23	23	23	23	23	23	23	23	23	23	21	13	10	10	10	10	10	10
1000	23	23	23	23	23	23	23	23	23	23	30	17	10	10	10	10	10	10
1200	23	23	23	23	23	23	23	23	23	23	40	21	10	10	10	10	10	10
1390	23	23	23	23	23	23	23	23	23	23	51	24	10	10	10	10	10	10
1590	18	18	18	18	18	18	18	18	18	18	58	25	9	9	9	9	9	9
1790	18	18	18	18	18	18	18	18	18	14	73	27	9	9	9	9	9	9
1990	18	18	18	18	18	18	18	18	18	22	80	30	9	9	9	9	9	9
2190	25	25	25	25	25	25	25	25	25	38	83	35	12	12	12	12	12	12
2390	25	25	25	25	25	25	25	25	25	49	83	36	12	12	12	12	12	12
2590	25	25	25	25	25	25	25	25	25	58	83	36	12	12	12	12	12	12
2790	25	25	25	25	25	25	25	25	25	73	83	85	13	12	12	12	12	12
2990	25	25	25	25	25	25	25	19	83	83	85	17	12	12	12	12	12	12
3190	25	25	25	25	25	25	25	27	83	83	85	21	12	12	12	12	12	12
3390	25	25	25	25	25	25	25	35	83	83	85	24	12	12	12	12	12	12
3590	25	25	25	25	25	25	25	45	83	83	85	26	12	12	12	12	12	12
3790	25	25	25	25	25	25	25	56	83	83	85	29	12	12	12	12	12	12
3990	25	25	25	25	25	25	25	69	83	83	85	31	12	12	12	12	12	12
4190	25	25	25	25	25	25	18	83	83	83	85	33	12	12	12	12	12	12
4330	25	25	25	25	25	25	23	83	83	83	85	35	12	12	12	12	12	12
4580	18	18	18	18	18	18	28	80	80	80	80	32	9	9	9	9	9	9
4780	23	23	23	23	23	23	41	76	76	76	76	32	10	10	10	10	10	10
4980	23	23	23	23	23	23	51	76	76	76	76	32	10	10	10	10	10	10
5180	23	23	23	23	23	23	63	76	76	76	76	32	10	10	10	10	10	10
5380	23	23	23	23	23	23	71	76	76	76	76	32	10	10	10	10	10	10
5580	23	23	23	23	23	21	76	76	76	76	76	32	10	10	10	10	10	10
5780	23	23	23	23	23	28	76	76	76	76	76	32	10	10	10	10	10	10
5980	23	23	23	23	23	36	76	76	76	76	76	32	10	10	10	10	10	10
6130	23	23	23	23	23	43	76	76	76	76	76	32	10	10	10	10	10	10
6330	18	18	18	18	18	49	73	73	73	73	71	32	10	10	10	10	10	10
6530	18	18	18	18	18	63	73	73	73	73	71	32	10	10	10	10	10	10
6730	27	27	27	27	21	78	78	78	78	78	74	32	10	10	10	10	10	10
6930	27	27	27	27	27	78	78	78	78	78	74	32	10	10	10	10	10	10
7130	27	27	27	27	33	78	78	78	78	78	74	32	10	10	10	10	10	10
7280	27	27	27	27	39	78	78	78	78	78	74	32	10	10	10	10	10	10
7480	20	20	20	20	41	73	73	73	73	73	71	32	10	10	10	10	10	10
7680	20	20	20	20	53	73	73	73	73	73	71	32	10	10	10	10	10	10
7780	20	20	20	20	63	73	73	73	73	73	71	32	10	10	10	10	10	10
7980	32	32	32	24	75	75	75	75	75	75	69	32	10	10	10	10	10	10
8180	32	32	32	28	75	75	75	75	75	75	69	32	10	10	10	10	10	10
8380	32	32	32	33	75	75	75	75	75	75	69	32	10	10	10	10	10	10
8580	32	32	32	38	75	75	75	75	75	75	69	32	10	10	10	10	10	10
8780	32	32	32	44	75	75	75	75	75	75	69	32	10	10	10	10	10	10
8980	32	32	32	51	75	75	75	75	75	75	69	32	10	10	10	10	10	10
9180	32	32	32	58	75	75	75	75	75	75	69	32	10	10	10	10	10	10
9380	32	32	32	63	75	75	75	75	75	75	69	32	10	10	10	10	10	10
9580	32	32	32	70	75	75	75	75	75	75	69	32	10	10	10	10	10	10
9705	32	32	24	75	75	75	75	75	75	75	69	32	10	10	10	10	10	10
9905	32	32	28	75	75	75	75	75	75	75	69	32	10	10	10	10	10	10
10105	32	32	33	75	75	75	75	75	75	75	69	32	10	10	10	10	10	10
10305	32	32	38	75	75	75	75	75	75	75	69	32	10	10	10	10	10	10
10505	32	32	45	75	75	75	75	75	75	75	69	32	10	10	10	10	10	10
10690	32	32	51	75	75	75	75	75	75	75	69	32	10	10	10	10	10	10
10890	32	32	58	75	75	75	75	75	75	75	69	32	10	10	10	10	10	10
11090	54	54	76	89	89	89	89	89	89	89	71	32	10	10	10	10	10	10
11290	54	54	80	89	89	89	89	89	89	89	71	32	10	10	10	10	10	10
11490	54	54	84	89	89	89	89	89	89	89	71	32	10	10	10	10	10	10
11690	54	54	89	89	89	89	89	89	89	89	71	32	10	10	10	10	10	10
12090	31	31	43	43	43	43	43	43	43	43	35	35	20	12	12	12	12	12
12290	54	54	27	27	27	27	27	27	27	27	52	52	26	15	15	15	15	15
12490	54	54	27	27	27	27	27	27	27	27	52	52	26	15	15	15	15	15
12690	54	54	27	27	27	27	27	27	27	27	52	52	26	15	15	15	15	15
12900	54	54	27	27	27	27	27	27	27	27	52	52	26	15	15	15	15	15
13100	28	28	19	19	19	19	19	19	19	19	28	28	19	12	12	12	12	12
13300	28	28	19	19	19	19	19	19	19	19	28	28	19	12	12	12	12	12
13500	28	28	19	19	19	19	19	19	19	19	28	28	19	12	12	12	12	12
13715	28	28	19	19	19	19	19	19	19	19	28	28	19	12	12	12	12	12
13915	28	28	19	19	19	19	19	19	19	19	28	28	19	12	12	12	12	12
14115	30	30	22	22	22	22	22	22	22	22	17	17	14	8	8	8	8	8

KRONOS																		
hypothetical test case																		
1-A																		
DENSITY in vehicles/mile/lane																		
DIST ft.	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
7180	24	24	24	24	24	24	24	24	24	24	12	12	12	12	12	12	12	12
7280	24	24	24	24	24	24	24	24	24	24	26	12	12	12	12	12	12	12
7380	24	24	24	24	24	24	24	24	24	36	12	12	12	12	12	12	12	12
7480	24	24	24	24	24	24	24	24	24	28	50	12	12	12	12	12	12	12
7580	24	24	24	24	24	24	24	24	24	38	63	12	12	12	12	12	12	12
7680	24	24	24	24	24	24	24	24	24	54	73	12	12	12	12	12	12	12
7780	26	26	26	26	26	26	26	26	43	86	86	13	13	13	13	13	13	13
7880	26	26	26	26	26	26	26	27	55	86	86	13	13	13	13	13	13	13
7980	26	26	26	26	26	26	32	65	86	86	23	13	13	13	13	13	13	13
8080	26	26	26	26	26	26	42	73	86	86	46	13	13	13	13	13	13	13
8180	26	26	26	26	26	27	54	78	86	86	67	13	13	13	13	13	13	13
8280	26	26	26	26	26	33	65	82	86	86	78	13	13	13	13	13	13	13
8380	26	26	26	26	26	44	73	84	86	86	83	13	13	13	13	13	13	13
8480	26	26	26	26	27	56	78	85	86	86	85	13	13	13	13	13	13	13
8580	26	26	26	26	34	66	82	86	86	86	86	13	13	13	13	13	13	13
8680	26	26	26	26	45	73	84	86	86	86	86	13	13	13	13	13	13	13
8780	26	26	26	28	57	79	85	86	86	86	86	13	13	13	13	13	13	13
8880	26	26	26	35	67	82	86	86	86	86	86	13	13	13	13	13	13	13
8980	26	26	26	47	74	84	86	86	86	86	86	13	13	13	13	13	13	13
9080	26	26	30	58	79	85	86	86	86	86	86	13	13	13	13	13	13	13
9180	26	26	38	68	82	86	86	86	86	86	86	13	13	13	13	13	13	13
9280	26	28	51	75	84	86	86	86	86	86	86	13	13	13	13	13	13	13
9380	26	36	61	80	85	86	86	86	86	86	86	13	13	13	13	13	13	13
9480	29	47	70	82	86	86	86	86	86	86	86	13	13	13	13	13	13	13
9580	36	58	76	84	86	86	86	86	86	86	86	13	13	13	13	13	13	13
9680	48	68	80	85	86	86	86	86	86	86	86	13	13	13	13	13	13	13
9805	59	74	83	86	86	86	86	86	86	86	86	17	13	13	13	13	13	13
9905	68	79	84	86	86	86	86	86	86	86	86	31	13	13	13	13	13	13
10005	38	38	38	38	38	38	38	38	38	38	38	15	15	15	15	15	15	15
10105	38	38	38	38	38	38	38	38	38	38	38	15	15	15	15	15	15	15
10205	37	37	37	37	37	37	37	37	37	37	37	15	15	15	15	15	15	15
10305	37	37	37	37	37	37	37	37	37	37	37	15	15	15	15	15	15	15
10405	37	37	37	37	37	37	37	37	37	37	37	15	15	15	15	15	15	15
10505	36	36	36	36	36	36	36	36	36	36	36	15	15	15	15	15	15	15
10605	36	36	36	36	36	36	36	36	36	36	36	15	15	15	15	15	15	15
10705	34	34	34	34	34	34	34	34	34	34	34	15	15	15	15	15	15	15
10805	33	33	33	33	33	33	33	33	33	33	33	15	15	15	15	15	15	15
10990	33	33	33	33	33	33	33	33	33	33	33	15	15	15	15	15	15	15
11090	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
11190	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
11290	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
11390	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
11490	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
11590	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
11690	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
11790	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
11890	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
11990	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
12090	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
12190	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
12290	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
12390	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
12490	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
12590	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
12690	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
12800	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
12900	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
13000	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
13100	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
13200	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
13300	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
13400	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15
13500	32	32	32	32	32	32	32	32	32	32	32	15	15	15	15	15	15	15

INTEGRATION

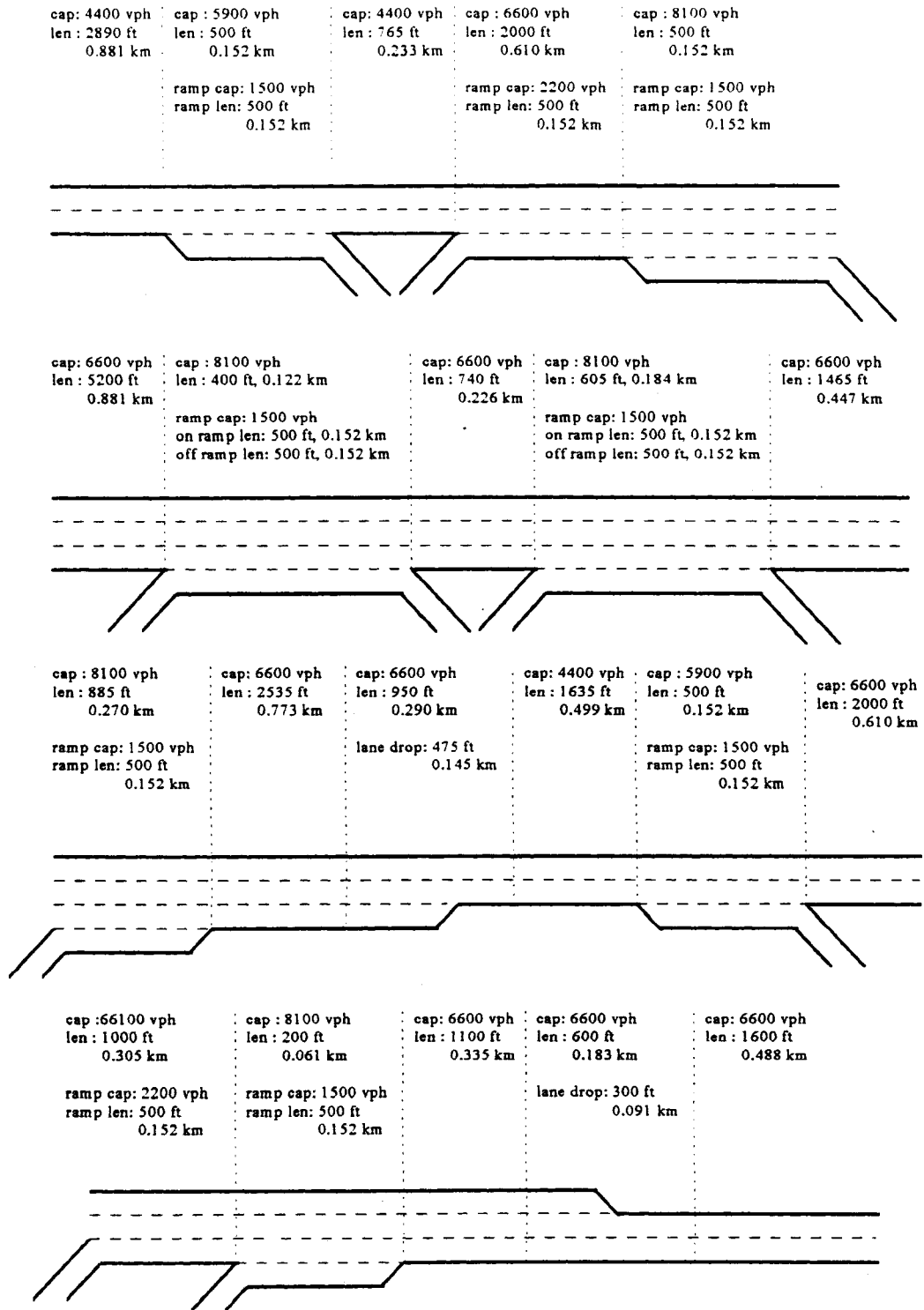
hypothetical test case  
1-A

DENSITY in vehicles/mile/lane

DIST ft	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
200	25	27	27	26	26	27	26	41	125	105	112	106	97	82	83	53	16	13
400	25	25	25	25	25	26	25	46	151	126	132	139	112	109	111	89	18	13
600	25	25	25	25	25	26	25	51	151	125	129	134	111	113	109	106	24	14
800	25	25	25	25	25	25	25	48	152	128	125	134	104	112	110	105	32	14
1000	25	24	25	25	24	25	26	45	146	126	124	131	97	114	110	103	38	15
1200	25	23	25	24	25	25	29	45	136	123	124	126	94	113	108	105	49	16
1390	25	23	25	24	24	24	33	80	136	126	126	128	90	115	114	107	70	18
1590	20	18	20	19	20	19	33	120	133	124	116	127	98	113	106	104	91	19
1790	21	19	20	20	20	20	40	137	134	126	128	139	100	121	121	120	114	25
1990	23	20	21	21	22	20	41	139	113	115	114	130	100	114	109	123	107	34
2190	28	25	27	26	27	27	51	140	127	126	131	130	92	101	95	96	89	54
2390	27	25	27	26	27	29	52	143	125	128	128	132	92	101	96	95	88	70
2590	27	25	27	26	27	29	62	136	122	128	129	130	94	98	94	93	88	88
2790	28	25	27	26	26	30	74	129	119	127	135	125	94	96	95	92	88	93
2990	28	25	26	26	26	38	101	126	125	127	131	126	97	93	96	92	89	91
3190	27	25	26	26	26	43	133	131	122	129	133	125	97	94	96	91	90	88
3390	28	25	26	26	26	50	149	119	120	127	131	120	96	99	96	93	89	90
3590	28	25	26	26	27	50	142	120	124	127	132	116	98	101	97	98	90	90
3790	28	25	26	26	30	49	144	119	121	124	126	122	99	106	98	104	92	94
3990	28	25	26	26	31	55	135	119	119	129	127	116	102	110	100	109	96	97
4190	29	26	26	26	33	80	124	120	116	132	131	112	107	112	96	114	100	98
4330	30	26	26	27	38	114	125	113	122	139	132	107	110	112	96	116	107	104
4580	24	21	21	22	41	97	93	95	90	96	98	77	79	82	73	84	84	74
4780	26	24	24	24	52	144	124	121	122	127	141	99	109	111	108	121	111	112
4980	26	25	24	24	54	141	133	120	120	124	132	93	110	111	118	122	114	115
5180	26	24	24	25	66	137	125	125	122	120	134	90	110	115	116	121	112	112
5380	26	24	24	27	90	137	135	125	123	121	132	90	114	114	117	121	110	110
5580	26	24	24	28	134	129	132	128	123	120	126	90	116	113	115	118	108	110
5780	26	24	24	31	141	134	134	126	123	130	126	91	118	111	113	113	109	110
5980	26	24	24	44	132	134	126	127	135	127	131	91	116	108	111	108	108	111
6130	25	24	24	57	130	140	125	120	131	129	119	95	116	110	105	107	108	112
6330	20	20	21	57	104	113	106	98	114	104	104	77	91	86	80	85	81	88
6530	21	21	21	86	105	113	115	110	118	122	125	87	101	90	89	89	91	96
6680	27	26	30	150	121	133	120	116	139	127	115	103	104	98	96	99	97	106
6880	27	27	33	144	122	130	120	116	133	128	112	105	104	99	97	99	99	107
7080	27	27	42	133	125	127	124	123	135	133	113	102	103	96	97	97	97	109
7280	28	28	60	131	125	129	121	122	134	130	112	100	101	94	96	97	96	110
7530	23	25	73	100	104	102	101	100	106	103	90	80	80	72	76	74	75	85
7780	23	29	93	102	114	120	109	107	119	113	106	87	85	78	81	82	83	91
7905	29	50	124	118	119	121	122	108	129	131	102	96	99	94	96	94	95	98
8105	31	63	124	118	121	124	124	111	133	131	102	99	99	94	97	96	97	100
8305	33	89	123	118	120	118	126	110	131	127	100	99	98	95	99	100	98	101
8505	37	109	122	113	121	120	128	113	128	134	96	100	100	96	100	102	99	101
8705	44	108	121	115	119	117	125	118	126	133	97	101	101	99	102	103	99	103
8905	57	109	122	116	119	116	126	118	119	133	96	102	103	102	105	106	100	106
9105	72	108	123	117	123	115	121	115	120	134	91	100	103	105	108	105	100	108
9305	85	107	122	123	124	112	126	117	114	131	88	99	104	108	114	106	102	110
9505	91	102	120	130	124	111	121	118	111	131	85	99	106	110	116	106	101	108
9705	90	98	121	126	129	109	122	119	112	125	84	98	106	112	118	104	102	111
9895	90	96	118	126	133	112	121	117	116	115	82	96	107	113	118	103	102	113
10095	91	93	120	125	142	111	119	122	116	111	83	94	107	115	118	101	101	113
10295	91	94	119	128	144	114	114	120	127	104	83	94	105	112	117	102	102	111
10495	96	97	123	125	145	118	117	124	136	105	84	96	104	109	114	105	104	111
10695	105	102	135	129	147	129	119	130	145	109	90	94	101	101	110	102	105	113
10895	88	85	123	99	126	116	96	105	125	88	79	81	83	81	86	87	83	88
11095	34	47	79	99	123	115	87	107	135	73	51	49	38	35	36	35	37	29
11295	37	49	78	122	130	113	96	115	129	98	71	59	42	38	39	35	39	30
11495	40	49	77	126	125	106	101	112	123	119	81	64	49	46	45	36	43	32
11695	43	53	81	123	121	99	103	113	116	118	82	69	55	54	54	40	50	37
11895	37	42	79	95	98	79	105	98	98	99	62	51	48	44	43	38	45	34
12095	39	41	72	84	85	69	87	78	78	78	52	47	53	43	40	41	45	33
12305	49	53	53	60	72	48	26	29	25	25	30	28	29	28	29	30	29	27
12505	48	57	63	75	83	54	28	28	25	24	28	27	28	27	27	28	28	26
12705	48	59	66	84	89	60	37	27	25	24	28	26	28	26	26	28	27	26
12905	49	62	69	84	100	69	45	28	25	24	27	26	28	26	26	28	27	26
13105	42	55	63	70	89	67	38	25	20	17	20	18	21	18	19	21	21	20
13305	53	75	98	92	110	93	55	31	24	18	21	18	23	19	20	22	21	21
13520	68	108	112	96	109	98	87	41	29	20	22	20	25	24	23	22	24	22
13720	106	124	112	100	102	102	103	59	36	25	25	25	38	38	29	26	35	26
13920	86	86	87	80	80	75	75	60	37	33	31	38	43	50	38	33	43	35
14120	22	23	23	26	24	25	25	24	25	23	22	21	20	23	22	22	21	23



## I-35W Southbound Through Burnsville, Minnesota



Flow Data for Test Case 2-A : I-35W Congestion Spillback

time slice	mainline vol. (vph)	ramp 1 vol. (vph)	ramp 2 vol. (vph)	ramp 3 vol. (vph)	ramp 4 vol. (vph)	ramp 5 vol. (vph)
1	3800	300	600	300	300	300
2	3800	300	600	300	300	300
3	3800	300	1200	300	300	300
4	3800	300	1200	300	300	300
5	3800	300	1200	300	300	300
6	3800	300	1200	300	300	300
7	3800	300	1200	300	300	300
8	3800	300	1200	300	300	300
9	3800	300	1200	300	300	300
10	3800	300	1200	300	300	300
11	1360	300	600	300	300	300
12	1360	300	600	300	300	300
13	1360	300	600	300	300	300
14	1360	300	600	300	300	300
15	1360	300	600	300	300	300
16	1360	300	600	300	300	300
17	1360	300	600	300	300	300
18	1360	300	600	300	300	300

time slice	ramp 6 vol. (vph)	ramp 7 vol. (vph)	ramp 8 vol. (vph)	ramp 9 vol. (vph)	ramp 10 vol. (vph)	ramp 11 vol. (vph)
1	300	300	300	300	300	300
2	300	300	300	300	300	300
3	300	300	600	300	600	300
4	300	300	600	300	600	300
5	300	300	600	300	600	300
6	300	300	600	300	600	300
7	300	300	600	300	600	300
8	300	300	600	300	600	300
9	300	300	600	300	600	300
10	300	300	600	300	600	300
11	300	300	300	300	300	300
12	300	300	300	300	300	300
13	300	300	300	300	300	300
14	300	300	300	300	300	300
15	300	300	300	300	300	300
16	300	300	300	300	300	300
17	300	300	300	300	300	300
18	300	300	300	300	300	300

## Calibration Parameters for Test Case 2-A : I-35W Congestion Spillback

### FRESIM :

- default vehicle set.
- car following factors = 15 to 6.
- acceleration lag = 0.3 sec.
- deceleration lag = 0.3 sec.
- lane change time = 3 sec.
- vehicle generation minimum separation = 0.5 sec.
- percentage of drivers willing to yield = 30 percent.
- minimum non-emergency deceleration = 0.8 ft/sec<sup>2</sup>

### AIMSUN2 :

#### vehicle characteristics:

- length = 4.0 m.
- width = 2.0 m.
- desired speed = 104 kph.
- maximum acceleration = 2.8 m/sec<sup>2</sup>
- maximum deceleration = 4.0 m/sec<sup>2</sup>

#### simulation parameters:

- simulation step size = 0.75sec
- speed acceptance = 0.80
- braking coefficient = 2.0
- zone 1 distance = 100 m
- zone 2 distance = 10 m
- minimum distance between vehicles = 1.5 m
- queue speed = 1.0 m/sec<sup>2</sup>
- queue leaving speed = 4.0 m/sec<sup>2</sup>
- junction visibility = 25 m
- cruising tolerance = 0.80

### FREFLO :

- relaxation time = 75 sec/mile
- anticipation coefficient = 0.25 miles<sup>2</sup>/hour
- capacities as listed on the freeway geometrics
- cubic polynomial speed-density relationship

$$\text{speed} = 81 - 125 (k * 10^{-2}) + 115 (k * 10^{-2})^2 - 49 (k * 10^{-2})^3$$

**FREQ11:**

- capacities as listed on the freeway geometrics
- points on the speed-V/C curve

Points on Upper Portion		of Speed-V/C curve	Points on Lower Portion		of Speed-V/C curve
V/C ratio	speed (mph)		V/C ratio	speed (mph)	
0.00	65		0.00	0	
0.44	65		0.16	6	
0.50	62		0.73	19	
0.73	53		0.86	24	
0.86	46		1.00	34	
1.00	34				

**KRONOS:**

- jam density = 136 vpm
- maximum capacity = 2200 vph
- density at maximum capacity = 68
- density at end of linear region of QK curve = 15 vpm
- flow rate at end of linear region of QK curve = 975 vph
- capacities as listed on the freeway geometrics

**INTEGRATION:**

- mainline freeflow speed = 104 kph
- ramp freeflow speed = 72 kph
- capacities as listed on the freeway geometrics
- speed at capacity for mainline = 54.7 kph
- speed at capacity for ramps = 54.7 kph
- mainline jam density = 84.5 vpkpl
- ramp jam density = 84.5 vpkpl

Flow Data for Test Case 2-B : I-35W Incident Spillback

time slice	mainline vol. (vph)	ramp 1 vol. (vph)	ramp 2 vol. (vph)	ramp 3 vol. (vph)	ramp 4 vol. (vph)	ramp 5 vol. (vph)
1	3800	300	600	300	300	300
2	3800	300	600	300	300	300
3	3800	300	600	300	300	300
4	3800	300	600	300	300	300
5	3800	300	600	300	300	300
6	3800	300	600	300	300	300
7	3800	300	600	300	300	300
8	3800	300	600	300	300	300
9	1360	300	600	300	300	300
10	1360	300	600	300	300	300
11	1360	300	600	300	300	300
12	1360	300	600	300	300	300
13	1360	300	600	300	300	300
14	1360	300	600	300	300	300
15	1360	300	600	300	300	300
16	1360	300	600	300	300	300
17	1360	300	600	300	300	300
18	1360	300	600	300	300	300

time slice	ramp 6 vol. (vph)	ramp 7 vol. (vph)	ramp 8 vol. (vph)	ramp 9 vol. (vph)	ramp 10 vol. (vph)	ramp 11 vol. (vph)
1	300	300	300	300	300	300
2	300	300	300	300	300	300
3	300	300	300	300	300	300
4	300	300	300	300	300	300
5	300	300	300	300	300	300
6	300	300	300	300	300	300
7	300	300	300	300	300	300
8	300	300	300	300	300	300
9	300	300	300	300	300	300
10	300	300	300	300	300	300
11	300	300	300	300	300	300
12	300	300	300	300	300	300
13	300	300	300	300	300	300
14	300	300	300	300	300	300
15	300	300	300	300	300	300
16	300	300	300	300	300	300
17	300	300	300	300	300	300
18	300	300	300	300	300	300

## Calibration Parameters for Test Case 2-B : I-35W Incident Spillback

### FRESIM :

- default vehicle set.
- car following factors = 15 to 6.
- acceleration lag = 0.3 sec.
- deceleration lag = 0.3 sec.
- lane change time = 3 sec.
- vehicle generation minimum separation = 0.5 sec.
- percentage of drivers willing to yield = 30 percent.
- minimum non-emergency deceleration = 0.8 ft/sec<sup>2</sup>

### AIMSUN2 :

#### vehicle characteristics:

- length = 4.0 m.
- width = 2.0 m.
- desired speed = 104 kph.
- maximum acceleration = 2.8 m/sec<sup>2</sup>
- maximum deceleration = 4.0 m/sec<sup>2</sup>

#### simulation parameters:

- simulation step size = 0.75sec
- speed acceptance = 0.80
- braking coefficient = 2.0
- zone 1 distance = 100 m
- zone 2 distance = 10 m
- minimum distance between vehicles = 1.5 m
- queue speed = 1.0 m/sec<sup>2</sup>
- queue leaving speed = 4.0 m/sec<sup>2</sup>
- junction visibility = 25 m
- cruising tolerance = 0.80

### FREFLO :

- relaxation time = 75 sec/mile
- anticipation coefficient = 0.25 miles<sup>2</sup>/hour
- capacities as listed on the freeway geometrics
- cubic polynomial speed-density relationship

$$\text{speed} = 81 - 125 (k * 10^{-2}) + 115 (k * 10^{-2})^2 - 49 (k * 10^{-2})^3$$

**FREQ11:**

- capacities as listed on the freeway geometrics
- points on the speed-V/C curve

Points on Upper Portion		of Speed-V/C curve	Points on Lower Portion		of Speed-V/C curve
V/C ratio	speed (mph)		V/C ratio	speed (mph)	
0.00	65		0.00	0	
0.44	65		0.16	6	
0.50	62		0.73	19	
0.73	53		0.86	24	
0.86	46		1.00	34	
1.00	34				

**KRONOS:**

- jam density = 136 vpm
- maximum capacity = 2200 vph
- density at maximum capacity = 68
- density at end of linear region of QK curve = 15 vpm
- flow rate at end of linear region of QK curve = 975 vph
- capacities as listed on the freeway geometrics

**INTEGRATION:**

- mainline freeflow speed = 104 kph
- ramp freeflow speed = 72 kph
- capacities as listed on the freeway geometrics
- speed at capacity for mainline = 54.7 kph
- speed at capacity for ramps = 54.7 kph
- mainline jam density = 84.5 vpkpl
- ramp jam density = 84.5 vpkpl

Density Tables for Test Case 2-A : I-35W SB Congestion Spillback

FRESIM																		
hypothetical test case 2-A																		
DENSITY in vehicles/mile/lane																		
DIST ft	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
13600	22	22	21	25	25	24	25	25	24	25	25	12	8	8	8	8	8	8
13850	20	20	20	24	23	22	23	24	22	23	23	10	6	5	6	5	6	6
14100	20	20	19	23	24	23	23	24	22	23	23	10	6	5	6	6	5	6
14350	20	20	20	23	24	22	23	24	22	23	23	11	6	6	6	6	5	5
14600	20	20	19	23	24	23	23	24	22	23	23	11	6	6	6	6	5	5
14850	19	20	19	23	24	23	23	24	22	23	23	12	6	5	6	6	5	5
15065	20	21	19	23	24	23	23	24	22	23	23	12	6	5	6	6	5	6
15315	22	23	23	27	28	28	27	28	26	27	25	15	7	8	8	8	7	7
15565	17	17	17	20	20	21	20	21	19	21	19	11	5	5	6	6	5	5
15815	16	17	16	20	20	21	20	21	19	20	19	11	5	6	5	5	5	5
16085	16	16	16	19	20	21	20	20	18	20	19	12	5	5	5	6	5	5
16335	21	22	22	25	27	27	26	27	25	28	29	18	7	7	7	8	7	7
16585	21	22	22	25	27	27	26	27	26	34	48	25	7	7	7	8	7	7
16835	21	22	22	25	27	26	26	28	32	49	67	43	7	7	7	7	7	7
17085	21	21	22	25	26	26	26	32	43	69	88	64	7	7	7	7	7	7
17335	21	21	22	25	26	26	27	49	67	85	96	82	9	7	7	7	7	7
17585	21	21	22	25	27	27	39	68	87	94	101	96	12	7	7	7	7	7
17835	22	22	23	25	28	37	68	83	97	100	104	101	20	7	7	7	7	7
18085	22	23	23	27	34	59	79	93	102	102	103	104	34	7	7	8	7	7
18335	23	23	23	27	40	69	90	99	105	104	104	105	47	7	7	7	7	7
18585	22	23	24	28	47	76	98	103	107	106	105	105	58	7	7	7	7	7
18835	24	24	25	32	57	86	99	101	103	102	99	100	61	7	7	7	7	7
19035	25	25	28	37	61	67	56	53	55	55	55	57	42	7	7	8	7	7
19285	37	37	40	46	54	51	51	49	51	51	51	51	42	11	11	11	10	10
19535	36	35	39	41	45	44	45	43	45	44	44	44	38	11	11	11	10	10
19785	35	34	37	40	43	42	43	41	42	41	42	41	36	11	11	11	11	11
20035	34	34	36	38	41	40	41	40	39	40	40	40	36	12	10	11	10	11
20285	34	33	36	38	41	40	40	40	39	39	40	40	36	12	10	12	10	11
20535	35	33	35	38	41	39	40	41	39	39	40	41	37	12	10	11	10	11
20785	34	34	35	37	41	40	39	41	40	39	40	41	37	12	11	11	10	10
21035	34	33	34	37	40	39	38	40	39	39	39	42	37	13	11	11	10	10
21320	33	33	33	36	38	38	37	38	38	38	40	41	38	13	11	11	11	10
21570	33	32	33	35	37	37	37	37	37	41	45	46	44	14	11	11	10	10
21820	31	30	30	32	34	34	34	35	34	46	56	56	53	14	9	9	8	9
22070	31	30	30	32	34	34	35	34	36	52	67	71	68	18	8	9	9	8
22320	31	30	31	32	34	33	35	35	39	58	76	78	78	24	9	9	8	9
22570	32	30	31	33	34	34	35	35	44	63	81	81	82	30	8	9	9	9
22820	32	31	31	33	34	33	36	37	50	69	85	81	83	36	8	9	9	9
23070	33	31	32	33	34	34	36	39	58	74	86	81	83	42	8	9	9	9
23320	33	31	32	33	35	34	36	44	66	80	85	82	83	49	8	9	9	9
23570	33	32	32	33	36	34	37	51	74	84	85	82	83	54	8	9	9	9
23820	36	34	37	39	41	40	45	67	97	106	97	92	93	67	11	12	12	12
24070	23	22	24	25	27	26	34	62	93	97	93	90	90	72	9	8	8	8
24320	23	22	23	24	26	29	45	85	106	109	108	108	110	96	14	7	8	7
24570	23	22	24	24	28	40	69	103	109	111	110	112	114	107	24	8	8	7
24820	27	26	28	31	46	72	101	112	113	113	112	113	116	113	40	10	10	10
25070	28	28	32	40	60	89	103	106	105	105	105	105	106	110	55	10	10	9
25320	28	29	32	44	68	94	104	106	103	105	104	104	105	110	71	9	9	9
25570	29	33	35	51	77	99	105	107	104	106	104	106	105	107	77	10	9	9
25820	34	38	41	61	82	98	98	102	96	98	97	99	95	98	71	11	9	9
26070	33	35	36	43	48	46	45	43	43	45	45	46	44	44	38	11	9	9
26320	41	43	42	44	46	46	47	46	46	46	46	47	45	47	44	16	13	13
26570	39	40	39	40	42	41	44	43	43	42	43	43	42	43	40	16	13	13
26820	38	39	38	38	39	39	41	41	41	40	40	41	40	41	39	16	13	13
27070	37	38	36	38	38	38	40	40	40	39	39	40	39	39	39	16	13	13
27570	37	37	36	38	38	37	38	39	40	38	39	40	38	38	39	17	14	13
28070	35	35	35	38	37	36	36	37	39	36	37	38	36	37	38	17	14	13



AIMSUN2

hypothetical test case  
2-A

DENSITY in vehicles/mile/lane

DIST ft	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
13600	9	12	11	14	12	13	13	12	13	13	13	6	4	5	5	4	5	5
13865	11	14	13	17	15	16	16	15	15	16	14	7	5	5	5	5	5	5
14065	11	14	13	16	15	16	16	15	15	16	15	7	5	5	5	5	5	5
14265	11	14	12	16	15	16	16	15	15	16	18	9	5	5	5	5	5	5
14465	11	14	12	16	15	15	16	15	16	19	27	18	5	5	5	5	5	5
14665	10	14	12	16	15	15	16	15	18	26	45	33	5	5	5	5	5	5
14865	10	14	13	16	15	15	16	16	24	39	58	45	5	5	5	5	5	5
15065	10	14	12	16	15	15	17	18	33	54	64	57	8	5	5	5	5	5
15265	9	12	12	15	14	14	16	22	43	67	81	73	23	5	5	5	5	5
15465	9	11	11	14	14	14	17	36	60	76	99	84	44	5	5	5	5	5
15665	8	11	11	14	15	15	29	59	69	97	116	97	62	11	5	4	5	4
15865	9	12	13	18	23	22	52	73	84	111	118	110	74	26	5	5	5	5
16085	12	17	18	26	34	35	65	79	87	100	103	101	75	38	7	7	7	7
16285	11	15	15	18	21	33	62	76	84	95	95	94	77	44	10	6	6	6
16485	11	15	15	17	21	42	66	80	89	96	94	94	84	51	16	6	6	6
16685	11	15	15	17	25	53	71	83	92	96	94	94	90	57	24	6	6	6
16885	11	14	15	17	31	62	75	86	94	96	93	94	93	64	33	6	6	6
17085	11	15	15	18	40	67	79	91	93	97	92	94	95	72	42	7	6	6
17285	11	14	15	20	51	71	84	94	94	96	92	95	94	80	49	13	6	6
17485	11	15	16	26	63	74	87	95	94	97	93	95	95	85	54	20	6	6
17685	11	16	20	39	70	79	90	95	95	96	93	95	94	91	58	29	6	6
17885	11	20	31	53	75	86	93	94	95	96	93	95	94	94	63	37	7	6
18085	13	31	48	65	82	94	94	95	96	95	94	95	94	95	70	42	9	6
18285	17	47	62	73	89	97	94	95	95	95	94	96	95	95	78	47	15	6
18485	24	65	75	80	93	97	95	95	97	95	95	96	96	95	84	53	23	6
18685	37	80	84	87	95	97	95	96	98	96	96	97	96	96	91	58	34	6
18885	55	92	94	94	98	99	97	97	100	98	98	98	98	98	97	68	45	11
19035	49	72	73	72	74	74	74	74	74	74	74	74	73	74	74	59	40	18
19235	21	30	30	30	30	30	30	30	31	30	31	30	31	31	31	26	19	12
19435	17	25	24	24	25	25	25	25	24	25	25	25	25	25	25	21	16	11
19635	16	23	22	22	23	23	22	24	22	23	23	23	23	23	23	19	15	10
19835	15	22	21	22	22	22	22	22	21	23	22	22	22	22	23	19	15	10
20035	15	21	21	21	21	21	21	22	21	22	22	21	22	21	22	18	15	10
20235	14	21	20	21	21	21	21	21	20	22	21	21	21	21	21	18	15	10
20435	14	21	21	21	21	20	21	21	20	21	21	21	21	21	21	18	14	10
20635	14	20	20	20	20	20	20	21	20	21	21	21	21	20	21	18	14	10
20835	14	20	20	20	20	20	20	21	20	21	22	21	21	21	21	18	15	11
20935	14	20	21	20	20	21	20	21	20	22	23	21	22	21	22	19	15	11
21170	11	17	17	17	17	17	17	18	17	19	21	20	19	18	19	17	12	9
21370	9	13	14	13	14	13	14	14	14	17	21	18	17	16	16	15	11	8
21570	9	14	14	14	14	14	14	15	16	23	31	24	21	20	21	19	13	10
21770	13	19	20	19	20	19	20	21	26	40	49	40	36	32	31	27	21	16
21970	13	19	19	19	19	19	19	22	31	46	57	54	47	44	37	32	27	19
22170	13	19	19	19	19	19	19	23	38	51	64	61	57	54	46	43	33	21
22370	13	19	19	19	19	19	20	26	44	57	69	67	64	63	57	55	37	25
22570	13	19	19	19	19	19	20	31	48	62	73	71	66	65	63	62	42	28
22770	13	19	19	19	19	19	22	36	52	65	76	74	69	66	66	64	48	35
22970	12	19	19	19	19	20	25	42	57	70	77	74	71	67	69	67	52	39
23170	12	19	19	19	19	20	31	45	63	75	77	75	72	71	72	71	56	40
23370	12	19	19	19	19	22	38	50	68	81	78	76	73	74	75	75	59	41
23570	12	19	19	20	20	25	45	57	75	85	79	76	75	75	76	78	63	44
23770	10	14	16	17	17	25	46	59	78	84	77	72	71	70	71	76	65	45
23970	9	13	16	17	18	37	62	83	99	101	101	100	98	99	101	102	93	62
24170	9	13	16	18	25	54	81	98	101	102	102	99	99	100	101	102	100	71
24370	9	14	17	22	43	79	95	102	101	102	102	100	98	100	101	103	102	84
24570	9	14	19	35	69	97	101	102	101	102	101	100	99	101	101	103	103	94
24770	9	15	24	52	81	94	96	97	101	100	93	87	86	90	94	98	102	96
24970	11	24	46	82	95	95	95	96	95	94	91	93	94	94	94	98	97	96
25170	13	41	67	92	95	95	96	97	95	94	92	95	94	95	94	98	96	97
25370	19	71	89	94	96	95	96	97	95	94	93	95	95	96	95	97	97	96
25620	29	90	95	96	97	96	96	98	96	95	95	95	96	96	96	98	98	97
25870	45	98	97	98	99	98	97	100	97	97	97	97	98	98	99	99	101	98
26070	35	61	62	62	61	61	61	62	61	61	61	61	61	62	62	62	62	61
26270	16	26	26	26	26	26	26	26	26	26	26	27	26	27	26	26	25	26
26470	14	23	23	23	23	23	23	23	23	24	24	24	24	24	23	23	23	23
26670	14	22	22	22	22	22	22	22	21	22	22	23	22	23	22	22	21	22
26870	13	21	22	21	21	21	22	21	22	22	22	22	22	21	21	21	21	21
27070	13	20	21	21	21	21	21	20	21	21	21	21	21	21	21	20	20	21
27270	13	20	21	21	21	21	21	20	21	21	21	21	21	20	21	20	20	21
27470	13	20	21	21	21	20	21	20	21	21	21	21	21	20	21	20	20	20
27670	13	20	21	21	20	20	20	20	21	21	21	21	21	20	20	20	20	20
27870	12	20	20	21	20	20	20	20	21	21	21	21	21	20	20	20	20	20
28070	12	20	20	20	20	20	20	20	21	21	21	21	21	20	20	19	20	20

FREFLO																		
hypothetical test case																		
2-A																		
DENSITY in vehicles/mile/lane																		
DIST ft.	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
13600	21	21	21	22	25	26	26	26	26	26	20	17	9	7	7	7	7	7
13850	21	21	22	22	25	26	26	26	26	26	20	17	9	7	7	7	7	7
14100	21	21	21	22	25	26	26	26	26	26	20	17	10	7	7	7	7	7
14350	21	21	21	22	25	26	26	26	26	26	21	17	10	7	7	7	7	7
14565	21	21	21	22	24	26	26	26	26	26	21	17	10	7	7	7	7	7
14815	17	17	18	19	21	22	22	22	22	22	17	14	9	7	7	7	7	7
15065	16	16	18	18	20	21	21	21	22	21	22	17	14	9	6	6	6	6
15315	16	16	18	18	20	21	21	21	21	21	18	14	10	6	6	6	6	6
15450	16	16	18	19	20	22	22	22	22	22	27	14	10	7	6	6	6	6
15715	23	23	24	26	28	30	31	31	30	34	61	23	14	9	9	9	9	9
15980	23	23	25	26	28	31	31	31	32	52	87	41	14	9	8	9	8	9
16245	23	23	25	27	28	32	32	32	35	78	90	75	17	8	8	9	8	9
16510	23	23	25	27	28	32	32	32	46	91	90	94	27	9	9	8	8	8
16775	23	23	25	27	28	32	32	33	69	91	94	93	47	9	8	8	8	8
17040	23	24	25	27	28	32	32	38	86	86	93	87	68	9	8	8	8	8
17305	23	24	25	27	28	32	33	57	91	88	92	89	82	9	8	8	8	8
17570	23	24	24	27	28	31	36	80	89	91	89	92	87	17	8	8	8	8
17835	23	24	24	27	28	31	47	95	86	93	87	94	88	29	8	8	8	8
18085	23	24	24	27	28	32	71	92	88	93	89	92	90	40	8	8	8	8
18335	24	24	24	27	28	39	92	87	91	87	92	86	91	49	9	8	8	8
18460	25	26	26	31	34	62	97	91	94	89	95	90	96	60	9	9	9	9
18735	41	42	43	50	61	97	100	99	99	98	99	98	100	76	13	13	13	13
19010	43	45	45	53	59	69	71	71	71	71	71	71	71	62	13	13	13	13
19285	44	46	46	53	58	62	64	64	64	63	64	64	64	60	13	13	13	13
19535	45	47	47	53	58	59	60	61	61	60	60	61	61	59	14	13	13	13
19785	45	47	48	53	57	58	59	59	59	58	59	59	59	59	16	13	13	13
20035	44	47	47	51	55	56	57	57	57	56	57	58	58	57	18	13	13	13
20285	43	46	46	49	53	53	54	54	54	54	55	57	56	54	20	13	13	12
20570	39	42	42	44	47	47	47	48	48	48	52	54	51	49	21	13	13	13
20820	24	26	26	27	29	29	30	30	30	32	39	37	31	30	14	8	8	8
21070	24	26	27	28	30	30	31	31	31	41	56	49	32	30	16	9	8	8
21320	35	38	38	40	44	44	44	45	45	64	88	78	45	41	21	10	10	10
21570	35	39	39	41	45	46	46	46	46	62	78	75	54	52	22	10	10	10
21820	35	39	40	41	46	46	47	47	47	64	76	73	64	64	24	10	10	10
22070	35	39	41	41	46	47	47	48	48	68	71	74	71	66	26	10	10	10
22320	35	39	41	41	46	47	47	48	48	75	71	74	70	69	29	10	10	10
22570	34	39	41	41	45	47	47	47	48	78	71	73	69	70	35	10	10	10
22820	34	39	40	40	44	46	46	46	49	81	72	71	70	68	52	10	10	10
23070	32	36	38	38	41	43	43	43	51	79	73	69	72	65	70	10	10	10
23320	22	25	28	28	30	31	32	33	56	74	66	64	67	56	69	9	9	9
23570	21	24	27	28	29	30	31	37	80	82	82	81	82	74	88	11	9	8
23820	20	23	26	27	28	30	31	51	96	84	93	92	94	89	89	27	9	9
24070	20	23	25	26	27	29	38	74	98	90	100	92	96	93	91	50	9	8
24290	16	18	20	21	22	25	54	95	95	98	101	91	98	94	95	61	8	8
24565	21	25	27	29	30	42	88	110	105	107	109	101	107	106	106	89	11	10
24840	21	25	27	29	32	60	96	100	97	99	97	94	96	99	97	91	17	10
25090	21	25	28	32	42	85	102	97	98	98	98	96	98	97	98	96	27	10
25390	22	27	31	46	77	102	100	102	102	101	101	101	102	101	100	103	41	10
25640	35	44	52	88	112	104	103	107	107	107	104	108	104	107	107	104	60	15
25890	36	46	53	68	74	72	70	72	73	74	72	72	72	73	73	71	49	15
26140	36	46	53	61	65	64	62	64	65	65	64	64	64	64	65	63	49	15
26390	35	47	54	59	61	61	59	60	62	61	61	61	61	61	61	60	50	15
26640	35	46	53	58	60	60	58	59	60	60	60	59	59	60	60	59	52	15
27140	33	45	53	57	59	59	58	58	59	59	59	58	59	58	59	58	55	16
27640	32	44	52	56	59	59	57	57	59	59	60	58	59	59	59	58	57	18

FREQ11

hypothetical test case  
2-A

DENSITY in vehicles/mile/lane

DIST ft.	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
13600	17	17	19	19	19	19	19	19	19	19	8	8	8	8	8	8	8	8
13865	21	21	24	24	24	24	24	24	24	24	10	10	10	10	10	10	10	10
14065	21	21	24	24	24	24	24	24	24	24	10	10	10	10	10	10	10	10
14265	21	21	24	24	24	24	24	24	24	24	10	10	10	10	10	10	10	10
14465	21	21	24	24	24	24	24	24	24	24	10	10	10	10	10	10	10	10
14665	21	21	24	24	24	24	24	24	24	24	10	10	10	10	10	10	10	10
14865	21	21	24	24	24	24	24	24	24	24	10	10	10	10	10	10	10	10
15065	21	21	24	24	24	24	24	24	24	20	11	10	10	10	10	10	10	10
15350	17	17	21	21	21	21	21	21	21	30	17	8	8	8	8	8	8	8
15550	17	17	21	21	21	21	21	21	21	46	26	8	8	8	8	8	8	8
15750	17	17	21	21	21	21	21	21	21	64	30	8	8	8	8	8	8	8
15950	17	17	21	21	21	21	21	21	21	37	74	34	8	8	8	8	8	8
16118	24	24	30	30	30	30	30	30	30	72	80	42	11	11	11	11	11	11
16285	24	24	30	30	30	30	30	30	36	85	80	83	11	11	11	11	11	11
16535	24	24	30	30	30	30	30	30	60	85	80	83	16	11	11	11	11	11
16785	24	24	30	30	30	30	34	80	85	80	83	24	11	11	11	11	11	11
17035	24	24	30	30	30	30	63	85	85	80	83	30	11	11	11	11	11	11
17285	24	24	30	30	30	36	80	85	85	80	83	36	11	11	11	11	11	11
17535	24	24	30	30	30	66	85	85	85	80	83	40	11	11	11	11	11	11
17785	24	24	30	30	38	85	85	85	85	80	83	78	11	11	11	11	11	11
18035	24	24	30	30	66	85	85	85	85	80	83	83	16	11	11	11	11	11
18285	24	24	30	41	85	85	85	85	85	80	83	83	24	11	11	11	11	11
18510	24	24	30	69	85	85	85	85	85	80	83	83	30	11	11	11	11	11
18735	24	24	42	85	85	85	85	85	85	80	83	83	34	11	11	11	11	11
18960	24	24	69	85	85	85	85	85	85	80	83	83	38	11	11	11	11	11
19185	53	53	64	64	64	64	64	64	64	64	71	74	74	37	17	17	17	17
19410	53	53	64	64	64	64	64	64	64	64	71	74	74	39	17	17	17	17
19645	53	53	64	64	64	64	64	64	64	64	74	74	74	17	17	17	17	17
19870	53	53	64	64	64	64	64	64	64	56	74	74	74	20	17	17	17	17
20070	53	53	64	64	64	64	64	64	64	56	74	74	74	22	17	17	17	17
20270	53	53	64	64	64	64	64	64	64	58	74	74	74	25	17	17	17	17
20470	53	53	64	64	64	64	64	64	64	58	74	74	74	27	17	17	17	17
20670	53	53	64	64	64	64	64	64	64	60	74	74	74	28	17	17	17	17
20870	53	53	64	64	64	64	64	64	64	60	74	74	74	30	17	17	17	17
21070	53	53	64	64	64	64	64	64	64	62	74	74	74	32	17	17	17	17
21370	25	25	28	28	28	28	28	28	28	49	75	74	74	33	11	11	11	11
21570	25	25	28	28	28	28	28	28	25	71	75	74	74	36	11	11	11	11
21820	41	41	48	48	48	48	48	48	49	80	80	79	79	40	15	15	15	15
22070	41	41	48	48	48	48	48	56	80	80	79	79	79	76	15	15	15	15
22320	41	41	48	48	48	48	48	63	80	80	79	79	79	79	19	15	15	15
22570	41	41	48	48	48	48	48	70	80	80	79	79	79	79	23	15	15	15
22820	41	41	48	48	48	48	42	80	80	80	79	79	79	79	26	15	15	15
23070	41	41	48	48	48	48	47	80	80	80	79	79	79	79	28	15	15	15
23320	41	41	48	48	48	48	55	80	80	80	79	79	79	79	31	15	15	15
23570	41	41	48	48	48	48	63	80	80	80	79	79	79	79	33	15	15	15
23770	24	24	28	28	28	28	72	80	80	80	80	80	80	80	38	12	12	12
23970	24	24	28	28	28	36	80	80	80	80	80	80	80	80	42	12	12	12
24170	24	24	28	28	28	57	80	80	80	80	80	80	80	80	80	12	12	12
24370	24	24	28	28	26	76	80	80	80	80	80	80	80	80	80	16	12	12
24570	24	24	28	28	43	80	80	80	80	80	80	80	80	80	80	22	12	12
24770	18	18	21	21	61	73	73	73	73	73	73	73	73	73	26	10	10	10
24970	27	27	31	39	81	81	81	81	81	81	81	81	81	81	81	36	13	13
25170	27	27	31	59	81	81	81	81	81	81	81	81	81	81	81	39	13	13
25370	27	27	29	77	81	81	81	81	81	81	81	81	81	81	81	43	13	13
25620	27	27	44	81	81	81	81	81	81	81	81	81	81	81	81	48	13	13
25870	27	27	70	81	81	81	81	81	81	81	81	81	81	81	81	81	40	13
26170	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	26	22
26370	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	26	22
26670	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	26	22
26970	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	26	22
27270	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	26	22
28070	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	26	22

KRONOS

hypothetical test case  
2-A

DENSITY in vehicles/mile/lane

DIST ft	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
15065	22	22	22	27	27	27	27	27	27	27	27	7	7	7	7	7	7	7
15165	21	22	22	27	27	27	27	27	27	27	27	7	7	7	7	7	7	7
15265	21	22	22	27	27	27	27	27	27	27	27	7	7	7	7	7	7	7
15365	21	22	22	27	27	27	27	27	27	27	27	7	7	7	7	7	7	7
15465	21	22	22	27	27	27	27	27	27	27	27	7	7	7	7	7	7	7
15565	21	22	22	27	27	27	27	27	27	27	27	7	7	7	7	7	7	7
15665	21	22	22	27	27	27	27	27	27	27	27	7	7	7	7	7	7	7
15765	21	22	22	27	27	27	27	27	27	27	27	7	7	7	7	7	7	7
15865	23	24	27	33	33	33	33	33	33	35	52	9	9	9	9	9	9	9
15965	23	24	27	33	33	33	33	33	33	37	70	9	9	9	9	9	9	9
16065	23	24	27	33	33	33	33	33	33	43	87	9	9	9	9	9	9	9
16250	23	24	27	33	33	33	33	33	33	55	99	10	9	9	9	9	9	9
16350	23	24	27	33	33	33	33	33	34	72	104	17	9	9	9	9	9	9
16450	23	24	27	33	33	33	33	33	35	88	106	40	9	9	9	9	9	9
16550	23	24	27	32	33	33	33	33	38	99	107	71	9	9	9	9	9	9
16650	23	24	27	32	33	33	33	33	46	104	107	94	9	9	9	9	9	9
16750	23	24	27	32	33	33	33	33	59	106	107	104	9	9	9	9	9	9
16850	23	24	27	32	33	33	33	34	76	107	107	107	9	9	9	9	9	9
16950	23	24	27	32	33	33	33	36	92	107	107	107	9	9	9	9	9	9
17050	24	24	27	32	33	33	33	40	101	107	107	107	9	9	9	9	9	9
17150	24	24	27	32	33	33	33	49	105	107	107	107	9	9	9	9	9	9
17250	24	24	27	32	33	33	34	63	107	107	107	107	9	9	9	9	9	9
17350	24	24	27	31	33	33	34	81	107	107	107	107	9	9	9	9	9	9
17450	24	24	27	31	33	33	36	95	107	107	107	107	9	9	9	9	9	9
17550	24	24	27	31	33	33	41	102	107	107	107	107	9	9	9	9	9	9
17650	24	24	27	31	33	33	52	108	107	107	107	107	9	9	9	9	9	9
17750	24	24	27	31	33	34	68	107	107	107	107	107	9	9	9	9	9	9
17850	24	24	27	31	33	35	85	107	107	107	107	107	9	9	9	9	9	9
17950	24	24	27	31	33	37	97	107	107	107	107	107	14	9	9	9	9	9
18050	24	24	27	30	33	43	104	107	107	107	107	107	32	9	9	9	9	9
18150	24	24	27	30	33	55	106	107	107	107	107	107	62	9	9	9	9	9
18250	24	24	27	30	33	72	107	107	107	107	107	107	89	9	9	9	9	9
18350	24	24	27	30	33	89	107	107	107	107	107	107	102	9	9	9	9	9
18450	24	24	27	30	33	100	107	107	107	107	107	107	106	9	9	9	9	9
18585	24	24	27	30	33	105	107	107	107	107	107	107	107	9	9	9	9	9
18685	24	24	27	30	33	106	107	107	107	107	107	107	107	9	9	9	9	9
18785	24	24	27	29	101	107	107	107	107	107	107	107	107	9	9	9	9	9
18885	26	26	29	32	108	108	108	108	108	108	108	108	108	9	9	9	9	9
18985	29	28	32	35	109	109	109	109	109	109	109	109	109	9	9	9	9	9
19085	32	31	35	38	108	108	108	108	108	108	108	108	108	10	10	10	10	10
19185	35	34	39	63	104	104	104	104	104	104	104	104	104	11	11	11	11	11
19285	50	48	59	63	64	65	66	66	66	66	66	67	67	13	13	13	13	13
19385	51	47	59	63	64	65	65	66	66	66	66	66	67	13	13	13	13	13
19535	51	47	58	63	64	65	65	66	66	66	66	66	67	13	13	13	13	13
19635	51	47	57	62	64	65	65	66	66	66	66	66	67	13	13	13	13	13
19735	51	47	56	62	64	65	65	66	66	66	66	66	67	13	13	13	13	13
19835	51	47	56	62	64	65	65	66	66	66	66	66	67	13	13	13	13	13
19935	51	47	55	61	63	64	65	65	66	66	66	66	67	13	13	13	13	13
20035	51	46	54	61	63	64	65	65	66	66	66	66	66	13	13	13	13	13
20135	51	47	54	61	63	64	65	65	66	66	66	66	66	13	13	13	13	13
20235	51	47	53	61	63	64	65	65	66	66	66	66	66	13	13	13	13	13
20335	51	47	53	60	63	64	65	65	66	66	66	66	67	14	13	13	13	13
20435	51	47	52	60	63	64	65	65	66	66	66	66	67	18	13	13	13	13
20535	51	47	52	60	63	64	65	65	66	66	66	66	67	27	13	13	13	13
20635	51	47	51	60	63	64	65	65	66	66	66	66	67	43	13	13	13	13
20735	51	47	51	60	62	64	65	65	66	66	66	66	67	60	13	13	13	13
20870	51	47	51	59	62	64	65	65	66	66	66	66	68	72	13	13	13	13
20970	51	47	51	59	62	64	65	65	66	66	66	66	69	78	13	13	13	13
21070	51	48	50	58	61	63	64	65	65	65	66	66	82	93	13	13	13	13
21170	50	48	50	58	61	63	64	65	65	65	66	66	83	93	13	13	13	13
21270	50	48	49	58	61	63	64	65	65	65	66	66	84	93	13	13	13	13
21370	50	48	49	57	61	63	64	65	65	65	66	66	86	93	13	13	13	13
21470	50	48	49	57	61	63	64	65	65	65	66	66	88	93	13	13	13	13
21570	50	48	49	57	61	63	64	65	65	65	66	66	89	93	13	13	13	13
21670	42	41	41	46	48	49	49	49	49	49	49	49	54	92	93	11	11	11
21770	42	41	41	46	48	49	49	49	49	49	49	49	50	57	92	93	11	11
21870	42	41	41	46	48	49	49	49	49	49	49	49	50	60	93	93	11	11
21970	42	41	41	46	48	49	49	49	49	49	49	49	50	66	93	93	11	11

KRONOS

hypothetical test case  
2-A

DENSITY in vehicles/mile/lane

DIST ft.	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
22070	42	42	41	46	48	49	49	49	49	49	51	72	93	93	11	11	11	11
22170	42	42	41	46	48	49	49	49	49	49	52	78	93	93	11	11	11	11
22270	42	42	41	46	48	49	49	49	49	49	53	83	93	93	11	11	11	11
22370	42	42	41	46	48	49	49	49	49	49	56	87	93	93	11	11	11	11
22470	42	42	41	45	48	49	49	49	49	49	60	90	93	93	11	11	11	11
22570	42	42	41	45	48	49	49	49	49	49	50	65	91	93	93	11	11	11
22670	42	42	41	45	48	49	49	49	49	49	50	72	92	93	93	11	11	11
22770	42	42	41	45	48	48	49	49	49	49	50	78	93	93	93	11	11	11
22870	42	42	41	45	48	48	49	49	49	49	51	83	93	93	93	11	11	11
22970	42	42	41	45	47	48	49	49	49	49	52	87	93	93	93	11	11	11
23070	42	42	40	45	47	48	49	49	49	49	55	90	93	93	93	11	11	11
23170	42	42	40	44	47	48	48	49	49	49	58	91	93	93	93	11	11	11
23270	42	42	40	44	47	48	48	48	48	48	64	92	93	93	93	17	11	11
23370	41	42	40	44	46	47	48	48	48	48	71	93	93	93	93	32	11	11
23470	41	41	39	43	45	46	47	47	47	47	79	93	93	93	93	54	11	11
23570	40	40	38	42	44	45	45	45	46	46	86	93	93	93	93	75	11	11
23670	37	37	36	39	41	41	42	42	44	44	91	93	93	93	93	86	10	10
23770	34	35	34	36	38	39	39	39	44	44	97	97	97	97	97	95	10	10
23870	32	32	32	34	36	36	37	37	49	49	103	102	102	102	102	102	10	10
23970	30	30	31	33	34	35	35	36	60	60	107	106	106	106	106	106	10	10
24070	28	29	29	31	32	33	34	36	77	77	109	108	108	108	108	108	9	9
24170	27	27	28	30	31	32	33	40	93	93	110	109	109	109	109	109	9	9
24270	26	26	27	29	30	31	35	49	103	110	110	110	110	110	110	110	9	9
24370	24	24	26	28	29	29	40	65	107	110	110	110	110	110	110	8	8	8
24470	24	25	26	28	29	30	54	84	109	110	110	110	110	110	110	9	9	9
24570	24	25	26	28	29	30	72	97	110	110	110	110	110	110	110	9	9	9
24670	24	25	26	28	29	30	90	104	110	110	110	110	110	110	110	9	9	9
24770	27	27	29	31	32	47	107	107	107	107	107	107	107	107	107	10	10	10
24870	27	27	29	30	33	60	107	107	107	107	107	107	107	107	107	107	13	10
24970	27	27	29	30	34	78	107	107	107	107	107	107	107	107	107	107	26	10
25070	27	27	29	30	36	93	107	107	107	107	107	107	107	107	107	107	53	10
25170	27	27	29	31	42	101	107	107	107	107	107	107	107	107	107	107	82	10
25270	27	27	29	31	55	105	107	107	107	107	107	107	107	107	107	107	99	10
25370	27	27	29	32	72	107	107	107	107	107	107	107	107	107	107	107	105	10
25470	27	27	29	36	89	107	107	107	107	107	107	107	107	107	107	107	107	10
25570	27	27	29	45	100	107	107	107	107	107	107	107	107	107	107	107	107	10
25670	27	27	29	61	105	107	107	107	107	107	107	107	107	107	107	107	107	10
25770	27	27	29	80	106	107	107	107	107	107	107	107	107	107	107	107	107	10
25870	27	27	66	107	107	107	107	107	107	107	107	107	107	107	107	107	107	10
25970	30	35	104	104	104	104	104	104	104	104	104	104	104	104	104	104	104	11
26070	62	66	66	67	67	67	67	67	67	67	67	67	67	67	67	67	67	15
26170	62	66	66	67	67	67	67	67	67	67	67	67	67	67	67	67	67	15
26270	63	66	66	67	67	67	67	67	67	67	67	67	67	67	67	67	67	15
26370	64	66	66	66	67	67	67	67	67	67	67	67	67	67	67	67	67	15
26470	64	66	66	66	67	67	67	67	67	67	67	67	67	67	67	67	67	15
26570	65	66	66	66	67	67	67	67	67	67	67	67	67	67	67	67	67	15
26670	66	66	66	66	67	67	67	67	67	67	67	67	67	67	67	67	67	15
26770	66	66	66	66	67	67	67	67	67	67	67	67	67	67	67	67	67	15
26870	67	66	66	66	67	67	67	67	67	67	67	67	67	67	67	67	67	15
26970	67	66	66	66	66	67	67	67	67	67	67	67	67	67	67	67	67	15

INTEGRATION																		
hypothetical test case																		
2-A																		
DENSITY in vehicles/mile/lane																		
DIST	TIMES/ICE																	
ft	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
13610	22	25	25	28	28	30	29	28	26	34	36	20	9	9	9	9	10	8
13810	23	23	25	26	28	29	30	30	28	28	29	17	7	7	7	8	7	7
14010	23	23	24	26	27	28	29	28	28	27	28	17	7	7	7	7	7	7
14210	23	22	24	25	27	28	28	28	28	27	28	18	7	7	7	7	7	7
14410	23	22	24	25	27	28	28	27	28	27	27	18	7	7	7	7	7	7
14610	23	22	23	25	27	27	27	27	28	26	27	18	7	7	7	7	7	7
14810	23	22	23	25	26	27	27	27	28	27	27	19	7	7	7	7	7	7
15010	22	21	23	25	25	26	26	26	27	26	26	19	7	7	7	7	7	7
15210	18	17	19	21	22	22	22	22	23	22	21	16	7	6	6	7	7	7
15410	18	17	19	21	22	22	22	22	22	22	21	16	7	7	6	7	7	7
15610	19	18	21	23	25	24	24	25	24	25	23	17	7	7	7	7	7	7
15895	20	20	25	28	28	34	29	29	30	32	29	26	7	7	7	8	7	7
16030	25	24	28	31	33	34	34	33	34	35	45	40	9	9	9	9	9	9
16230	26	24	28	31	34	34	34	33	34	41	59	81	10	9	9	10	9	9
16430	26	24	28	30	33	34	34	33	34	51	79	112	11	9	9	9	9	9
16630	26	24	28	30	33	33	34	33	39	76	97	114	13	9	9	9	9	9
16830	26	24	27	30	33	33	34	34	58	102	107	115	15	9	9	9	9	9
17030	26	24	27	29	33	33	34	38	92	106	109	116	18	9	9	9	9	9
17230	26	24	27	29	33	33	35	53	109	106	110	116	24	9	9	9	9	9
17430	26	24	27	29	33	33	39	71	109	107	110	115	39	9	9	9	9	9
17630	26	24	27	29	33	33	50	95	108	107	110	115	84	9	9	9	9	9
17830	25	24	27	30	33	35	75	105	109	106	111	114	111	10	9	9	9	9
18030	26	24	27	29	33	43	98	106	109	108	111	114	112	12	9	9	10	9
18230	26	24	27	29	37	74	107	106	107	107	113	115	111	14	9	9	9	9
18430	25	25	27	30	48	103	109	106	108	109	116	116	114	16	9	9	9	9
18630	32	29	32	45	88	107	110	110	111	110	116	124	120	21	10	9	10	10
18905	47	46	61	79	114	117	115	117	115	125	129	125	121	37	11	10	11	10
19040	48	47	51	58	63	65	65	65	65	66	73	79	80	37	14	14	15	14
19175	47	45	49	56	62	64	64	64	64	64	74	80	82	42	14	14	15	13
19375	47	46	49	56	61	63	65	65	65	66	77	80	83	55	14	14	15	14
19575	47	46	48	55	59	62	65	65	66	66	78	80	82	67	14	14	15	13
19775	46	46	48	55	59	63	65	66	66	67	80	79	81	79	15	13	15	13
19975	45	47	47	54	59	63	65	66	67	67	81	79	80	83	16	14	15	13
20175	45	47	46	54	59	63	66	67	66	67	83	81	80	83	17	14	15	13
20375	44	47	46	53	59	63	66	68	67	68	84	82	79	85	18	14	15	14
20575	43	47	45	53	59	64	66	68	67	69	85	82	80	88	20	14	15	13
20775	43	47	45	52	60	66	67	68	67	71	85	82	82	91	21	14	14	13
21010	43	47	45	52	62	68	66	67	65	73	85	84	85	94	24	14	15	14
21115	28	30	28	33	42	46	42	42	42	48	56	57	57	68	15	9	9	9
21315	31	34	30	34	45	48	44	43	43	48	58	57	58	72	18	9	9	9
21515	31	33	31	34	42	44	42	39	41	43	56	57	51	61	19	11	10	9
21715	37	41	40	44	46	47	53	50	50	60	86	81	75	43	22	11	12	11
21915	36	40	40	43	45	45	50	48	49	65	90	90	90	50	22	11	12	11
22115	36	40	40	43	45	45	49	48	49	77	88	95	91	63	23	11	11	11
22315	36	39	40	41	44	45	49	47	50	93	89	95	90	81	24	11	12	11
22515	35	39	40	41	44	46	49	47	52	102	87	94	90	90	26	11	11	11
22715	35	39	39	40	44	46	49	47	57	101	85	94	89	92	29	11	11	11
22915	35	39	39	40	44	47	48	48	62	99	85	92	89	93	34	11	11	11
23115	34	39	38	40	43	47	48	48	69	97	83	92	89	94	43	11	11	11
23315	33	39	38	40	43	46	47	48	78	97	83	90	91	92	61	11	11	11
23515	32	37	37	38	42	44	44	47	88	99	83	88	93	91	79	11	11	11
23715	22	26	27	28	30	32	32	37	82	87	73	74	79	78	74	10	9	9
23915	22	25	27	28	29	31	31	54	106	109	104	106	110	105	105	12	9	9
24115	22	24	26	27	29	31	32	100	108	112	110	110	110	108	110	14	9	9
24315	22	24	26	28	29	30	41	114	109	112	109	110	110	107	108	16	9	9
24515	21	24	26	27	29	32	75	113	110	112	110	110	108	106	107	19	9	9
24715	20	23	23	24	27	37	86	103	92	97	97	91	87	89	87	22	9	9
24915	24	27	28	30	33	72	107	107	108	109	107	108	110	109	108	41	11	11
25115	23	27	28	30	42	101	109	109	108	110	109	109	110	110	110	68	11	11
25315	23	27	30	37	72	108	109	107	107	110	109	108	109	109	109	106	12	11
25565	29	32	46	70	103	107	106	106	107	106	106	106	105	106	106	105	15	11
25815	32	51	87	110	116	111	110	112	109	107	108	107	108	108	108	110	19	12
26015	42	52	61	63	64	65	64	65	65	64	64	64	64	64	65	65	26	17
26215	42	49	58	60	62	62	63	62	62	62	62	62	62	62	62	62	27	17
26415	42	49	57	58	61	62	62	63	63	63	63	63	63	63	63	63	28	17
26615	41	49	56	59	61	62	63	63	63	63	63	63	63	63	63	63	29	17
26815	41	49	55	59	61	62	63	63	63	63	63	63	63	63	63	63	30	17

Density Tables for Test Case 2-B : I-35W Incident Spillback

FRESIM																		
hypothetical test case																		
2-B																		
DENSITY in vehicles/mile/lane																		
DIST	TIMESLICE																	
ft	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
16085	16	16	15	15	16	16	16	16	16	11	6	5	5	5	5	5	5	6
16335	21	22	20	20	21	21	21	21	21	15	7	7	7	7	7	7	6	8
16585	21	22	20	20	22	20	21	21	21	15	8	7	7	7	7	7	6	8
16835	21	22	20	20	21	20	21	21	21	17	8	7	7	7	6	7	6	8
17085	21	21	20	20	21	20	21	21	21	31	12	7	7	7	7	7	6	7
17335	21	21	21	20	22	20	21	21	24	56	34	7	8	7	6	7	6	8
17585	21	21	21	20	22	20	21	21	32	98	80	13	8	7	6	7	6	7
17835	22	22	21	21	22	20	21	21	46	114	119	30	8	7	6	7	6	7
18085	22	23	22	22	23	21	23	26	80	121	124	50	8	7	7	7	6	8
18335	23	23	22	22	24	22	22	34	88	127	123	64	9	7	6	7	6	8
18585	22	23	23	23	24	22	23	38	96	130	127	73	10	7	7	7	6	8
18835	24	24	24	24	26	23	24	50	105	144	132	76	10	7	7	8	6	8
19035	25	25	27	26	28	24	25	61	104	122	110	65	10	7	7	8	7	7
19285	37	37	39	38	39	37	37	104	126	128	110	71	15	11	10	11	10	11
19535	36	35	36	35	37	35	37	118	127	125	107	71	16	10	10	11	9	11
19785	35	34	35	34	35	34	46	131	126	124	104	71	17	10	10	11	9	11
20035	34	34	35	32	34	34	54	119	125	126	101	70	19	10	10	11	10	11
20285	34	33	35	32	34	33	65	121	126	128	98	69	20	10	10	10	10	11
20535	35	33	35	33	34	33	92	126	125	126	95	67	21	10	10	10	10	11
20785	34	34	35	33	34	34	93	121	127	123	93	64	22	11	10	11	10	11
21035	34	33	35	33	34	38	113	122	125	124	90	61	24	11	10	11	10	11
21320	33	33	33	32	32	48	113	133	140	140	95	59	25	11	10	11	10	11
21570	33	32	33	32	32	65	149	181	188	184	127	68	30	10	10	11	9	11
21820	31	30	30	30	29	79	147	147	150	143	98	78	36	8	8	8	8	9
22070	31	30	30	30	30	93	145	144	144	135	92	82	45	8	8	8	8	9
22320	31	30	30	30	32	115	143	146	146	133	89	84	52	8	8	8	8	9
22570	32	30	31	30	38	120	144	144	146	129	89	83	58	8	8	8	9	8
22820	32	31	32	30	44	135	143	144	146	125	89	83	63	8	8	8	8	8
23070	33	31	32	30	60	139	144	144	148	120	88	83	67	9	8	8	9	8
23320	33	31	32	31	72	139	142	145	146	117	87	83	71	11	8	8	8	8
23570	33	32	32	32	88	139	148	143	145	113	86	83	75	14	8	8	8	8
23820	36	34	35	37	115	179	184	188	186	131	96	93	84	21	11	11	11	11
24070	23	22	22	32	105	143	149	148	150	116	92	89	85	22	7	7	7	7
24320	23	22	22	49	136	153	154	154	156	123	113	108	107	39	7	7	7	7
24570	23	22	22	84	155	156	153	153	155	122	118	113	112	61	6	7	7	7
24820	27	26	34	127	158	157	157	159	158	117	118	116	115	80	10	10	10	9
25070	28	28	52	118	149	148	151	148	148	110	108	108	108	90	11	9	9	8
25320	28	29	51	124	148	150	148	148	144	105	105	105	104	96	17	9	9	8
25570	29	33	57	135	151	149	149	151	139	107	106	105	107	98	21	9	9	8
25820	34	38	78	145	150	154	151	149	139	103	101	98	95	93	21	9	9	8
26070	33	35	62	90	87	86	89	86	71	46	45	44	42	45	18	9	9	8
26320	41	43	94	92	69	61	59	56	46	43	45	46	46	46	25	13	12	12
26570	39	40	28	24	23	24	23	23	33	40	42	43	42	43	25	13	13	12
26820	38	39	23	20	19	19	19	19	28	39	41	41	40	40	25	13	12	12
27070	37	38	22	18	17	17	17	17	26	38	40	40	40	39	25	13	13	12
27570	37	37	22	16	16	16	16	16	25	37	39	40	40	38	26	13	13	12
28070	35	35	22	16	16	16	16	16	23	35	37	38	37	36	27	12	13	12

AIMSUN2																		
hypothetical test case																		
2-B																		
DENSITY in vehicles/mile/lane																		
DIST ft	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
16085	12	17	16	16	19	18	16	15	17	9	6	6	7	7	6	7	6	6
16285	11	15	14	15	16	16	16	16	17	9	6	6	6	6	6	6	6	6
16485	11	15	14	14	16	16	17	19	24	11	6	6	6	6	5	6	6	6
16685	11	15	14	14	15	17	21	25	29	15	6	6	6	6	5	6	6	6
16885	11	14	14	15	16	20	29	33	36	20	6	6	6	6	5	6	6	6
17085	11	15	14	14	18	27	35	45	46	28	6	6	6	6	5	6	6	6
17285	11	14	14	15	23	33	47	64	59	38	9	6	6	6	5	6	6	6
17485	11	15	15	17	32	42	64	69	67	54	17	6	6	6	5	6	6	6
17685	11	16	19	23	44	52	70	69	68	65	30	10	6	6	5	6	6	6
17885	11	19	30	40	56	62	72	72	71	70	60	23	7	6	5	6	6	6
18085	12	30	49	61	66	68	75	79	80	77	85	63	17	6	5	6	6	6
18285	16	50	65	68	70	71	83	85	89	85	87	86	43	6	5	6	6	6
18485	23	69	75	72	78	80	89	91	93	97	88	88	59	6	5	6	6	6
18685	36	83	86	83	89	88	96	96	94	106	100	89	69	10	5	6	6	6
18885	54	94	96	95	97	97	100	98	97	110	123	104	73	22	7	8	8	8
19035	49	73	73	74	73	73	74	73	73	94	115	103	61	30	12	14	14	14
19235	20	29	30	30	30	30	30	30	32	75	100	91	36	17	9	10	11	10
19435	17	24	25	24	24	24	24	25	31	84	101	89	37	15	9	10	9	9
19635	16	22	23	22	23	22	22	23	35	90	101	86	38	15	8	9	9	9
19835	15	21	22	21	22	22	21	22	43	95	100	83	40	17	8	9	9	9
20035	15	20	22	20	21	21	20	22	53	98	100	80	41	19	8	9	9	9
20235	14	20	21	20	21	21	20	23	63	101	99	75	42	21	8	9	9	9
20435	14	20	21	20	21	21	20	25	74	102	100	71	43	23	8	9	9	9
20635	14	20	21	20	20	21	20	29	84	101	99	68	43	26	8	9	9	9
20835	14	20	21	20	20	20	20	36	92	101	98	65	44	28	8	9	9	9
20935	14	20	21	20	21	21	20	45	96	102	98	64	45	30	8	9	9	9
21170	12	16	17	16	17	17	18	51	95	98	93	53	39	27	7	7	7	7
21370	9	13	14	13	14	14	17	82	126	127	119	65	42	30	6	6	6	6
21570	9	14	14	13	14	15	27	113	126	126	115	78	60	46	9	6	6	6
21770	13	19	19	19	19	20	55	133	136	138	113	76	70	55	14	7	7	8
21970	13	19	20	19	19	21	75	134	137	139	109	78	76	59	20	7	7	7
22170	13	18	19	19	19	23	97	134	137	138	103	77	78	61	26	7	7	8
22370	13	18	19	19	19	27	117	134	137	137	98	77	79	64	34	7	7	8
22570	13	18	19	19	19	36	131	134	137	135	93	77	79	68	41	7	7	7
22770	13	18	19	19	19	53	135	135	138	132	88	77	79	69	47	7	8	7
22970	12	18	19	19	20	71	135	135	139	130	84	76	80	69	52	9	7	8
23170	12	18	19	19	21	89	134	136	139	125	82	77	80	71	55	13	7	7
23370	12	19	19	19	24	106	135	137	140	120	81	78	80	75	58	18	8	7
23570	12	18	19	19	31	121	136	138	139	115	81	80	81	79	61	24	7	7
23770	10	13	14	15	37	133	136	137	138	112	78	79	78	77	63	31	6	6
23970	9	13	14	16	74	145	144	145	145	120	104	103	103	103	90	53	6	6
24170	9	13	14	19	111	144	143	143	143	115	104	104	102	103	96	68	12	6
24370	9	13	14	33	136	145	144	145	143	111	104	104	102	103	101	77	25	6
24570	9	14	17	67	145	145	145	146	143	108	104	104	103	103	103	85	43	6
24770	9	14	20	91	147	151	151	151	146	114	111	112	112	113	112	99	65	16
24970	11	23	42	132	149	149	148	149	137	96	93	96	95	95	94	92	70	28
25170	13	39	72	140	144	144	144	144	129	96	94	96	95	96	94	96	77	38
25370	18	68	97	142	143	143	143	144	124	97	94	96	96	96	95	97	85	47
25620	28	90	109	143	142	142	143	143	119	97	95	97	96	97	96	97	91	55
25870	44	100	116	146	146	146	146	146	115	99	97	98	98	99	98	99	98	68
26070	35	62	100	132	131	131	132	131	78	61	61	62	62	62	62	61	62	51
26270	16	25	102	127	127	127	128	127	45	26	26	26	26	26	26	26	26	23
26470	14	22	77	87	87	87	88	87	35	23	23	23	23	23	23	24	23	21
26670	14	21	14	14	14	14	14	14	27	22	22	22	22	22	22	22	22	19
26870	13	20	11	10	10	11	10	11	25	21	22	21	22	21	21	22	21	19
27070	13	20	10	9	9	9	9	10	24	21	22	21	21	21	21	21	21	19
27270	13	19	11	9	9	9	9	9	23	21	21	21	21	20	21	21	21	18
27470	13	19	10	9	9	9	8	9	23	20	21	21	20	20	21	20	20	19
27670	13	19	11	9	9	9	8	9	22	20	21	20	20	20	21	20	20	18
27870	13	19	11	9	8	8	8	9	21	20	21	20	20	20	21	20	20	18
28070	12	19	11	9	9	8	8	9	21	20	21	20	20	20	21	20	20	18



FREFLO

hypothetical test case  
2-B

DENSITY in vehicles/mile/lane

DIST ft.	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
12005	21	21	21	21	21	21	21	21	21	18	15	8	7	7	7	7	7	7
12255	21	21	21	21	21	21	21	21	21	18	15	8	7	7	7	7	7	7
12495	21	21	21	21	21	21	21	21	21	18	15	8	7	7	7	7	7	7
12695	17	17	17	17	17	17	17	17	17	15	13	9	7	7	7	7	7	7
12900	16	16	16	16	16	16	16	16	16	15	16	21	6	6	6	6	6	7
13100	16	16	16	16	16	16	16	16	16	15	34	25	7	6	6	6	6	7
13350	21	21	21	21	21	21	21	21	21	16	55	30	7	7	7	7	7	7
13600	21	21	21	21	21	21	21	21	21	17	87	30	7	7	7	7	7	7
13850	21	21	21	21	21	21	21	21	21	28	118	48	7	7	7	7	7	7
14100	21	21	21	21	21	21	21	21	21	54	125	78	13	7	7	7	7	7
14350	21	21	21	21	21	21	21	21	21	79	126	98	22	7	7	7	7	7
14565	21	21	21	21	21	21	21	21	21	103	131	96	32	7	7	7	7	7
14815	17	17	17	17	17	17	17	17	20	120	115	93	45	7	7	7	7	7
15065	16	16	16	16	16	16	16	16	40	133	120	105	71	7	6	6	6	6
15315	16	16	16	16	16	16	16	16	74	130	122	112	93	7	6	6	6	6
15450	16	16	17	16	17	17	16	114	138	138	135	99	18	6	6	6	6	6
15715	23	23	23	23	23	23	23	23	123	127	129	128	107	34	9	9	9	9
15980	23	23	23	23	23	23	23	27	125	120	113	109	93	42	9	8	9	8
16245	23	23	23	23	23	23	41	122	125	103	106	90	52	9	9	8	8	9
16510	23	23	23	24	24	23	58	120	128	101	102	94	62	8	8	8	8	9
16775	23	23	23	24	24	23	68	120	128	104	100	93	77	9	8	9	8	8
17040	23	24	23	24	24	24	87	120	128	104	101	90	87	14	8	8	8	8
17305	23	24	24	24	24	25	110	122	126	104	100	88	94	19	8	8	8	8
17570	23	24	24	24	24	36	124	121	124	103	101	87	94	31	8	8	8	8
17835	23	24	24	24	24	47	127	127	122	101	101	90	91	48	8	8	8	8
18085	23	24	24	24	24	58	129	129	127	99	100	94	88	59	8	8	9	8
18335	24	24	24	24	24	73	135	125	130	97	96	92	87	71	8	8	9	8
18460	25	26	26	26	26	91	145	153	148	95	103	91	92	79	9	9	9	9
18735	41	42	42	43	42	103	134	134	131	103	108	97	100	95	14	13	13	13
19010	43	45	45	45	45	101	125	117	119	77	79	71	71	71	17	13	13	13
19285	44	46	46	47	46	108	121	122	116	74	70	64	64	64	20	13	13	13
19535	45	47	47	48	48	110	121	120	118	77	64	61	61	61	23	13	13	13
19785	45	47	48	47	48	118	114	124	115	78	62	60	59	59	26	13	13	13
20035	44	47	47	47	51	115	116	119	112	81	61	59	58	57	28	13	13	13
20285	43	46	46	46	54	113	123	118	112	88	60	58	56	55	30	13	13	13
20570	39	42	42	42	54	111	120	118	112	89	58	56	53	51	29	13	13	13
20820	24	26	26	26	49	110	118	105	98	69	39	33	32	31	20	8	8	8
21070	24	26	26	27	68	120	127	117	106	83	53	34	32	30	20	8	9	9
21320	35	38	38	38	94	135	131	133	128	103	85	53	44	37	27	10	10	10
21570	35	39	39	39	90	121	116	119	112	77	77	63	51	38	28	10	10	10
21820	35	39	40	40	95	123	114	117	104	69	75	66	58	39	30	10	10	10
22070	35	39	41	40	100	120	111	122	99	68	71	70	66	44	32	10	10	10
22320	35	39	41	41	114	121	115	119	95	67	71	74	72	55	35	10	10	10
22570	34	39	41	45	121	116	117	117	95	68	72	78	71	69	39	10	10	10
22820	34	39	40	50	121	114	116	116	96	69	76	80	71	77	50	12	10	10
23070	32	36	37	52	117	119	116	115	95	70	82	82	72	78	71	15	10	10
23320	22	25	26	51	114	116	113	110	92	62	79	70	63	70	77	16	9	9
23570	21	24	25	67	116	121	125	120	100	78	94	83	79	84	91	29	9	9
23820	20	23	24	82	124	121	120	123	99	89	91	90	91	85	94	46	8	9
24070	20	23	24	99	122	124	123	120	97	99	88	94	97	90	95	62	9	8
24290	16	18	24	117	119	125	121	121	97	103	86	95	98	95	93	79	8	8
24565	21	25	44	129	125	131	128	128	107	111	101	106	104	106	107	102	13	10
24840	21	25	60	120	123	121	123	120	100	98	97	99	94	98	101	92	29	10
25090	21	25	78	116	121	116	123	122	99	99	97	98	95	98	100	96	49	10
25390	22	27	94	120	119	118	119	120	101	101	99	101	101	102	101	101	59	10
25640	35	44	114	122	123	116	123	119	108	104	105	107	107	108	103	106	80	15
25890	36	46	107	113	109	110	111	107	83	76	71	73	73	73	70	71	62	15
26140	36	46	57	56	56	56	55	55	79	71	63	65	65	65	63	63	59	15
26390	35	47	28	26	26	26	26	26	60	64	60	61	62	62	60	60	58	16
26640	35	46	25	21	21	21	21	21	52	61	59	60	60	60	59	59	58	19
27140	33	45	26	19	19	19	19	19	46	60	58	59	59	60	58	58	57	23
27640	32	44	27	18	18	18	18	18	42	59	58	59	59	59	58	58	57	26

FREQ11

hypothetical test case  
2-B

DENSITY in vehicles/mile/lane

DIST ft.	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
7155	21	21	21	21	21	21	21	21	21	21	8	8	8	8	8	8	8	8
7405	21	21	21	21	21	21	21	21	21	21	8	8	8	8	8	8	8	8
7655	21	21	21	21	21	21	21	10	70	21	8	8	8	8	8	8	8	8
7905	21	21	21	21	21	21	21	12	84	21	8	8	8	8	8	8	8	8
8155	21	21	21	21	21	21	21	15	84	21	8	8	8	8	8	8	8	8
8405	21	21	21	21	21	21	21	18	84	21	8	8	8	8	8	8	8	8
8655	21	21	21	21	21	21	21	21	84	21	8	8	8	8	8	8	8	8
8905	21	21	21	21	21	21	21	25	84	21	8	8	8	8	8	8	8	8
9155	21	21	21	21	21	21	21	30	84	21	8	8	8	8	8	8	8	8
9405	21	21	21	21	21	21	21	36	84	21	8	8	8	8	8	8	8	8
9655	21	21	21	21	21	21	21	41	84	21	8	8	8	8	8	8	8	8
9905	21	21	21	21	21	21	21	48	84	21	8	8	8	8	8	8	8	8
10155	21	21	21	21	21	21	21	59	84	21	8	8	8	8	8	8	8	8
10405	21	21	21	21	21	21	10	67	84	21	8	8	8	8	8	8	8	8
10655	21	21	21	21	21	21	12	67	84	21	8	8	8	8	8	8	8	8
10905	21	21	21	21	21	21	14	67	84	21	8	8	8	8	8	8	8	8
11155	21	21	21	21	21	21	17	67	84	21	8	8	8	8	8	8	8	8
11405	21	21	21	21	21	21	21	67	84	21	8	8	8	8	8	8	8	8
11655	21	21	21	21	21	21	24	67	84	21	8	8	8	8	8	8	8	8
11855	21	21	21	21	21	21	28	67	84	21	8	8	8	8	8	8	8	8
12055	17	17	17	17	17	17	32	59	73	17	7	7	7	7	7	7	7	7
12255	17	17	17	17	17	17	37	59	73	17	7	7	7	7	7	7	7	7
12395	21	21	21	21	21	21	44	67	84	21	9	9	9	9	9	9	9	9
12595	21	21	21	21	21	21	48	67	84	21	9	9	9	9	9	9	9	9
12795	21	21	21	21	21	21	59	67	84	21	9	9	9	9	9	9	9	9
12995	21	21	21	21	21	9	67	67	84	21	9	9	9	9	9	9	9	9
13195	17	17	17	17	17	10	59	59	73	17	8	8	8	8	8	8	8	8
13400	17	17	17	17	17	12	59	59	73	17	8	8	8	8	8	8	8	8
13600	17	17	17	17	17	15	59	59	73	17	8	8	8	8	8	8	8	8
13865	21	21	21	21	21	20	67	67	84	21	10	10	10	10	10	10	10	10
14065	21	21	21	21	21	23	67	67	84	21	10	10	10	10	10	10	10	10
14265	21	21	21	21	21	27	67	67	84	21	10	10	10	10	10	10	10	10
14465	21	21	21	21	21	31	67	67	84	21	10	10	10	10	10	10	10	10
14665	21	21	21	21	21	36	67	67	84	21	10	10	10	10	10	10	10	10
14865	21	21	21	21	21	41	67	67	84	21	10	10	10	10	10	10	10	10
15065	21	21	21	21	21	44	67	67	84	21	10	10	10	10	10	10	10	10
15350	17	17	17	17	17	53	59	59	73	17	8	8	8	8	8	8	8	8
15550	17	17	17	17	8	59	59	59	73	17	8	8	8	8	8	8	8	8
15750	17	17	17	17	11	59	59	59	73	17	8	8	8	8	8	8	8	8
15950	17	17	17	17	13	59	59	59	73	17	8	8	8	8	8	8	8	8
16118	24	24	24	24	20	70	70	70	80	24	11	11	11	11	11	11	11	11
16285	24	24	24	24	23	70	70	70	80	24	11	11	11	11	11	11	11	11
16535	24	24	24	24	25	70	70	70	80	24	11	11	11	11	11	11	11	11
16785	24	24	24	24	30	70	70	70	80	24	11	11	11	11	11	11	11	11
17035	24	24	24	24	35	70	70	70	80	24	11	11	11	11	11	11	11	11
17285	24	24	24	24	42	70	70	70	80	24	11	11	11	11	11	11	11	11
17535	24	24	24	24	49	70	70	70	80	24	11	11	11	11	11	11	11	11
17785	24	24	24	24	58	70	70	70	80	24	11	11	11	11	11	11	11	11
18035	24	24	24	24	63	70	70	70	80	24	11	11	11	11	11	11	11	11
18285	24	24	24	12	70	70	70	70	80	24	11	11	11	11	11	11	11	11
18510	24	24	24	15	70	70	70	70	80	24	11	11	11	11	11	11	11	11
18735	24	24	24	18	70	70	70	70	80	24	11	11	11	11	11	11	11	11
18960	24	24	24	21	70	70	70	70	80	24	11	11	11	11	11	11	11	11
19185	53	53	53	35	79	79	79	79	68	53	17	17	17	17	17	17	17	17
19410	53	53	53	37	79	79	79	79	68	53	17	17	17	17	17	17	17	17
19645	53	53	53	37	79	79	79	79	68	53	17	17	17	17	17	17	17	17
19870	53	53	53	38	79	79	79	79	68	53	17	17	17	17	17	17	17	17

FREQ11

hypothetical test case  
2-B

DENSITY in vehicles/mile/lane

DIST ft.	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
20070	53	53	53	40	79	79	79	79	68	53	17	17	17	17	17	17	17	17
20270	53	53	53	41	79	79	79	73	68	53	17	17	17	17	17	17	17	17
20470	53	53	53	41	79	79	79	79	68	53	17	17	17	17	17	17	17	17
20670	53	53	53	43	79	79	79	79	68	53	17	17	17	17	17	17	17	17
20870	53	53	53	45	79	79	79	79	68	53	17	17	17	17	17	17	17	17
21070	53	53	53	45	79	79	79	79	68	53	17	17	17	17	17	17	17	17
21370	25	25	25	37	63	63	63	63	76	25	11	11	11	11	11	11	11	11
21570	25	25	25	42	63	63	63	63	76	25	11	11	11	11	11	11	11	11
21820	41	41	41	53	73	73	73	73	79	41	15	15	15	15	15	15	15	15
22070	41	41	41	57	73	73	73	73	79	41	15	15	15	15	15	15	15	15
22320	41	41	41	62	73	73	73	73	79	41	15	15	15	15	15	15	15	15
22570	41	41	41	67	73	73	73	73	79	41	15	15	15	15	15	15	15	15
22820	41	41	41	73	73	73	73	73	79	41	15	15	15	15	15	15	15	15
23070	41	41	18	73	73	73	73	73	79	41	15	15	15	15	15	15	15	15
23320	41	41	19	73	73	73	73	73	79	41	15	15	15	15	15	15	15	15
23570	41	41	20	73	73	73	73	73	79	41	15	15	15	15	15	15	15	15
23770	24	24	15	70	70	70	70	70	80	24	12	12	12	12	12	12	12	12
23970	24	24	18	70	70	70	70	70	80	24	12	12	12	12	12	12	12	12
24170	24	24	21	70	70	70	70	70	80	24	12	12	12	12	12	12	12	12
24370	24	24	23	70	70	70	70	70	80	24	12	12	12	12	12	12	12	12
24570	24	24	28	70	70	70	70	70	80	24	12	12	12	12	12	12	12	12
24770	18	18	29	61	61	61	61	61	73	18	10	10	10	10	10	10	10	10
24970	27	27	41	73	73	73	73	73	82	27	13	13	13	13	13	13	13	13
25170	27	27	46	73	73	73	73	73	82	27	13	13	13	13	13	13	13	13
25370	27	27	49	73	73	73	73	73	82	27	13	13	13	13	13	13	13	13
25620	27	27	56	73	73	73	73	73	82	27	13	13	13	13	13	13	13	13
25870	27	27	67	73	73	73	73	73	82	27	13	13	13	13	13	13	13	13
26170	65	65	79	79	79	79	79	79	65	65	22	22	22	22	22	22	22	22
26370	65	65	32	32	32	32	32	32	65	65	22	22	22	22	22	22	22	22
26670	65	65	18	18	18	18	18	18	65	65	22	22	22	22	22	22	22	22
26970	65	65	18	18	18	18	18	18	65	65	22	22	22	22	22	22	22	22
27270	65	65	18	18	18	18	18	18	65	65	22	22	22	22	22	22	22	22
28070	65	65	18	18	18	18	18	18	65	65	22	22	22	22	22	22	22	22

KRONOS

hypothetical test case  
2-B

DENSITY in vehicles/mile/lane

DIST ft.	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
14500	21	22	22	22	22	22	22	22	22	23	7	7	7	7	7	7	7	7
14600	22	22	22	22	22	22	22	22	22	57	7	7	7	7	7	7	7	7
14700	21	22	22	22	22	22	22	22	22	95	7	7	7	7	7	7	7	7
14800	22	22	22	22	22	22	22	22	23	115	7	7	7	7	7	7	7	7
14965	22	22	22	22	22	22	22	22	27	120	7	7	7	7	7	7	7	7
15065	22	22	22	22	22	22	22	22	40	120	7	7	7	7	7	7	7	7
15165	21	22	22	22	22	22	22	22	69	120	7	7	7	7	7	7	7	7
15265	21	22	22	22	22	22	22	22	102	120	7	7	7	7	7	7	7	7
15365	21	22	22	22	22	22	22	22	121	120	7	7	7	7	7	7	7	7
15465	21	22	22	22	22	22	22	22	126	120	7	7	7	7	7	7	7	7
15565	21	22	22	22	22	22	22	22	127	120	7	7	7	7	7	7	7	7
15665	21	22	22	22	22	22	22	22	127	120	13	7	7	7	7	7	7	7
15765	21	22	22	22	22	22	22	22	127	119	28	7	7	7	7	7	7	7
15865	23	24	24	24	24	24	24	24	25	125	117	111	9	9	9	9	9	9
15965	23	24	24	24	24	24	24	24	26	125	117	111	9	9	9	9	9	9
16065	23	24	24	24	24	24	24	24	30	125	117	111	9	9	9	9	9	9
16250	23	24	24	24	24	24	24	24	43	125	117	111	9	9	9	9	9	9
16350	23	24	24	24	24	24	24	24	71	125	116	111	9	9	9	9	9	9
16450	23	24	24	24	24	24	24	101	125	116	111	9	9	9	9	9	9	9
16550	23	24	24	24	24	24	24	119	125	116	111	9	9	9	9	9	9	9
16650	23	24	24	24	24	24	24	124	125	116	111	9	9	9	9	9	9	9
16750	23	24	24	24	24	24	24	125	125	116	110	9	9	9	9	9	9	9
16850	23	24	24	24	24	24	24	125	125	116	110	9	9	9	9	9	9	9
16950	23	24	24	24	24	24	24	125	125	116	110	9	9	9	9	9	9	9
17050	24	24	24	24	24	24	24	125	125	115	110	12	9	9	9	9	9	9
17150	24	24	24	24	24	24	24	125	125	115	110	27	9	9	9	9	9	9
17250	24	24	24	24	24	24	24	25	125	115	110	56	9	9	9	9	9	9
17350	24	24	24	24	24	24	24	26	125	125	115	110	87	9	9	9	9	9
17450	24	24	24	24	24	24	24	33	125	125	115	110	103	9	9	9	9	9
17550	24	24	24	24	24	24	24	51	125	125	115	110	108	9	9	9	9	9
17650	24	24	24	24	24	24	24	81	125	125	115	110	110	9	9	9	9	9
17750	24	24	24	24	24	24	24	109	125	125	114	110	110	9	9	9	9	9
17850	24	24	24	24	24	24	24	122	125	125	114	110	110	9	9	9	9	9
17950	24	24	24	24	24	24	24	125	125	125	114	110	110	9	9	9	9	9
18050	24	24	24	24	24	24	24	125	125	125	114	110	110	9	9	9	9	9
18150	24	24	24	24	24	24	24	125	125	125	114	110	110	9	9	9	9	9
18250	24	24	24	24	24	24	24	125	125	125	114	110	110	9	9	9	9	9
18350	24	24	24	24	24	24	24	125	125	125	114	110	110	9	9	9	9	9
18450	24	24	24	24	24	24	24	125	125	125	114	110	110	9	9	9	9	9
18585	24	24	24	24	24	24	24	125	125	125	113	110	110	15	9	9	9	9
18685	24	24	24	24	24	24	24	125	125	125	113	110	110	28	9	9	9	9
18785	24	24	24	24	24	24	24	125	125	125	113	110	110	110	9	9	9	9
18885	26	26	26	26	26	31	130	130	130	130	115	112	112	112	9	9	9	9
18985	29	28	28	28	28	131	136	136	136	117	113	113	113	9	9	9	9	9
19085	32	31	31	31	31	141	141	141	141	119	113	113	113	10	10	10	10	10
19185	35	34	35	35	35	147	147	147	147	119	112	112	111	11	11	11	11	11
19285	50	48	49	49	49	119	119	119	119	94	87	86	86	13	13	13	13	13
19385	51	47	49	49	49	119	119	119	119	93	87	86	86	13	13	13	13	13
19535	51	47	49	49	49	119	119	119	119	93	87	86	86	13	13	13	13	13
19635	51	47	49	49	49	119	119	119	119	93	86	86	86	13	13	13	13	13
19735	51	47	49	49	49	119	119	119	119	92	86	86	86	13	13	13	13	13
19835	51	47	49	49	49	119	119	119	119	92	86	86	86	13	13	13	13	13
19935	51	47	49	49	49	119	119	119	119	92	86	86	86	13	13	13	13	13
20035	51	46	49	49	49	119	119	119	119	91	86	86	86	13	13	13	13	13
20135	51	47	49	49	49	119	119	119	119	91	86	86	86	13	13	13	13	13
20235	51	47	49	49	49	119	119	119	119	91	86	86	86	13	13	13	13	13
20335	51	47	49	49	49	119	119	119	119	90	86	86	86	13	13	13	13	13
20435	51	47	49	49	49	119	119	119	119	90	86	86	86	13	13	13	13	13
20535	51	47	49	49	49	119	119	119	119	90	86	86	86	13	13	13	13	13
20635	51	47	49	49	49	119	119	119	119	89	86	86	86	13	13	13	13	13
20735	51	47	49	49	49	119	119	119	119	89	86	86	86	13	13	13	13	13
20870	51	47	49	49	49	119	119	119	119	89	86	86	86	13	13	13	13	13
20970	51	47	49	49	49	119	119	119	119	89	86	86	86	13	13	13	13	13

KRONOS

hypothetical test case  
2-B

DENSITY in vehicles/mile/lane

DIST ft	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
21070	51	48	49	49	54	122	122	122	122	94	93	93	93	27	13	13	13	13
21170	50	48	49	49	60	122	122	122	122	94	93	93	93	49	13	13	13	13
21270	50	48	49	49	72	122	122	122	122	94	93	93	93	71	13	13	13	13
21370	50	48	49	49	89	122	122	122	121	94	93	93	93	85	13	13	13	13
21470	50	48	49	49	105	122	122	122	121	94	93	93	93	91	13	13	13	13
21570	50	48	49	49	114	122	122	122	120	94	93	93	93	92	13	13	13	13
21670	42	41	41	42	122	122	122	122	120	94	93	93	93	93	11	11	11	11
21770	42	41	41	42	122	122	122	122	119	94	93	93	93	93	11	11	11	11
21870	42	41	41	42	122	122	122	122	119	94	93	93	93	93	11	11	11	11
21970	42	41	41	42	122	122	122	122	118	93	93	93	93	93	11	11	11	11
22070	42	42	41	42	122	122	122	122	117	93	93	93	93	93	11	11	11	11
22170	42	42	41	42	122	122	122	122	116	93	93	93	93	93	11	11	11	11
22270	42	42	41	42	122	122	122	122	115	93	93	93	93	93	11	11	11	11
22370	42	42	41	42	122	122	122	122	115	93	93	93	93	93	11	11	11	11
22470	42	42	41	42	122	122	122	122	114	93	93	93	93	93	11	11	11	11
22570	42	42	41	42	122	122	122	122	113	93	93	93	93	93	11	11	11	11
22670	42	42	41	42	122	122	122	122	112	93	93	93	93	93	11	11	11	11
22770	42	42	41	42	122	122	122	122	111	93	93	93	93	93	11	11	11	11
22870	42	42	41	42	122	122	122	122	110	93	93	93	93	93	11	11	11	11
22970	42	42	41	42	122	122	122	122	109	93	93	93	93	93	11	11	11	11
23070	42	42	40	42	122	122	122	122	108	93	93	93	93	93	11	11	11	11
23170	42	42	40	42	122	122	122	122	107	93	93	93	93	93	11	11	11	11
23270	42	42	40	43	122	122	122	122	106	93	93	93	93	93	11	11	11	11
23370	41	42	40	45	122	122	122	122	105	93	93	93	93	93	11	11	11	11
23470	41	41	39	51	122	122	122	122	104	93	93	93	93	93	11	11	11	11
23570	40	40	38	62	122	122	122	122	103	93	93	93	93	93	11	11	11	11
23670	37	37	36	81	122	122	122	122	102	93	93	93	93	93	15	10	10	10
23770	34	35	33	102	124	124	124	124	105	97	97	97	97	97	32	10	10	10
23870	32	32	31	116	125	125	125	125	108	102	102	102	102	102	61	10	10	10
23970	30	30	29	123	125	125	125	125	109	106	106	106	106	106	88	10	10	10
24070	28	29	28	125	125	125	125	125	110	108	108	108	108	108	103	9	9	9
24170	27	27	26	125	125	125	125	125	110	109	109	109	109	109	108	9	9	9
24270	26	26	25	125	125	125	125	125	110	110	110	110	110	110	109	9	9	9
24370	24	24	23	125	125	125	125	125	110	110	110	110	110	110	110	8	8	8
24470	24	25	24	125	125	125	125	125	110	110	110	110	110	110	110	9	9	9
24570	24	25	24	125	125	125	125	125	110	110	110	110	110	110	110	9	9	9
24670	24	25	24	125	125	125	125	125	110	110	110	110	110	110	110	9	9	9
24770	27	27	58	124	124	124	124	124	107	107	107	107	107	107	107	10	10	10
24870	27	27	85	124	124	124	124	124	107	107	107	107	107	107	107	10	10	10
24970	27	27	108	124	124	124	124	124	107	107	107	107	107	107	107	10	10	10
25070	27	27	120	124	124	124	124	124	107	107	107	107	107	107	107	11	10	10
25170	27	27	123	124	124	124	124	124	107	107	107	107	107	107	107	16	10	10
25270	27	27	123	124	124	124	124	124	107	107	107	107	107	107	107	34	10	10
25370	27	27	124	124	124	124	124	124	107	107	107	107	107	107	107	64	10	10
25470	27	27	124	124	124	124	124	124	107	107	107	107	107	107	107	89	10	10
25570	27	27	124	124	124	124	124	124	107	107	107	107	107	107	107	102	10	10
25670	27	27	124	124	124	124	124	124	107	107	107	107	107	107	107	106	10	10
25770	27	27	124	124	124	124	124	124	107	107	107	107	107	107	107	107	10	10
25870	27	27	124	124	124	124	124	124	107	107	107	107	107	107	107	107	10	10
25970	30	35	127	127	127	127	127	127	104	104	104	104	104	104	104	104	11	11
26070	61	65	116	116	116	116	116	116	67	65	65	66	66	66	66	66	15	15
26170	62	65	116	116	116	116	116	116	67	65	65	65	66	66	66	66	15	15
26270	62	65	116	116	116	116	116	116	67	65	65	65	66	66	66	66	15	15
26370	63	65	116	116	116	116	116	116	66	65	65	65	66	66	66	66	15	15
26470	64	81	116	116	116	116	116	116	66	65	65	65	65	66	66	66	15	15
26570	64	65	67	67	67	68	68	34	65	64	65	65	65	66	66	66	15	15
26670	65	65	67	67	67	68	68	34	65	64	65	65	65	66	66	66	15	15
26770	65	48	18	18	18	18	18	18	63	64	65	65	65	65	66	66	15	15
26870	65	64	18	18	18	18	18	18	63	64	64	65	65	65	66	66	15	15
26970	65	64	18	18	18	18	18	18	61	64	64	65	65	65	66	66	15	15
27070	66	64	18	18	18	18	18	18	61	64	64	65	65	65	66	66	15	15
27170	65	64	18	18	18	18	18	18	59	63	64	65	65	65	66	66	15	15
27270	66	64	18	18	18	18	18	18	59	63	64	64	65	65	65	66	15	15
27370	65	64	18	18	18	18	18	18	57	63	64	64	65	65	65	66	15	15
27470	65	64	18	18	18	18	18	18	57	63	64	64	65	65	65	66	15	15
27570	65	64	18	18	18	18	18	18	55	62	63	64	65	65	65	66	15	15
27670	65	64	18	18	18	18	18	18	55	62	63	64	65	65	65	66	15	15
27770	64	64	18	18	18	18	18	18	52	61	63	64	65	65	65	65	16	15
27870	64	64	18	18	18	18	18	18	52	61	63	64	64	65	65	65	16	15
27970	64	64	18	18	18	18	18	18	50	61	63	64	64	65	65	65	17	15
28070	64	64	18	18	18	18	18	18	50	61	63	64	64	65	65	65	20	15

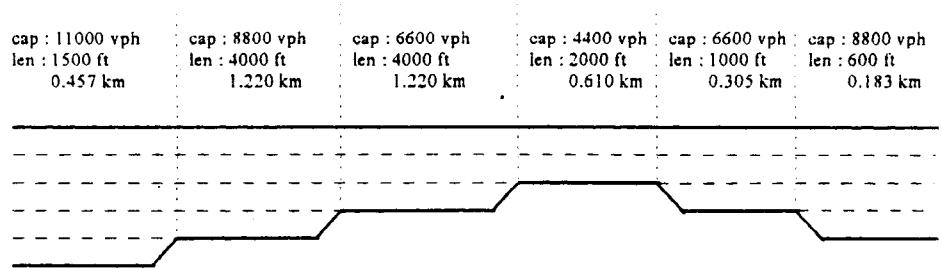
INTEGRATION

hypothetical test case  
2-B

DENSITY in vehicles/mile/lane

DIST ft	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
16030	25	24	26	25	26	23	26	25	27	23	9	9	9	9	9	9	9	9
16230	26	24	26	25	26	24	26	25	27	33	9	9	9	9	9	9	9	9
16430	26	24	26	25	26	24	26	25	27	60	10	9	9	9	9	9	9	9
16630	26	24	26	25	26	24	26	25	29	101	11	9	9	9	9	9	9	9
16830	26	24	26	25	26	24	25	25	40	103	14	9	9	9	9	9	9	9
17030	26	24	26	25	26	24	25	25	64	101	16	9	9	9	9	9	9	9
17230	26	24	26	25	25	24	25	25	117	101	20	9	9	9	9	9	9	9
17430	26	24	26	25	25	24	25	26	147	100	30	9	9	9	9	9	9	9
17630	26	24	26	25	25	24	25	27	142	101	51	9	9	9	9	9	9	9
17830	25	24	26	25	25	24	25	37	131	101	95	9	9	9	9	9	9	9
18030	26	24	26	25	25	24	25	65	123	100	111	11	9	9	9	9	9	9
18230	26	24	26	25	25	24	25	114	124	101	110	13	9	9	9	9	9	9
18430	25	25	26	25	25	25	25	149	124	104	109	15	9	9	9	9	9	9
18630	32	29	32	30	29	29	31	136	128	109	114	19	9	10	10	9	9	9
18905	47	46	48	48	49	41	62	131	130	106	133	38	10	11	10	11	11	11
19040	48	47	48	49	46	47	79	117	108	77	78	35	14	15	14	14	14	14
19175	47	45	48	49	44	46	90	118	109	79	80	40	14	14	14	14	14	14
19375	47	46	49	50	45	46	98	120	110	81	81	55	14	15	14	14	14	15
19575	47	46	47	50	45	47	105	120	109	81	81	65	14	15	14	14	14	14
19775	46	46	47	50	45	47	106	121	108	82	80	75	15	15	14	14	14	14
19975	45	47	46	49	45	47	116	122	105	83	80	77	16	15	14	14	14	14
20175	45	47	45	49	45	47	130	121	102	85	81	80	17	15	14	14	14	14
20375	44	47	45	49	45	48	136	121	99	86	82	80	18	14	14	14	14	14
20575	43	47	45	48	46	50	136	118	94	88	84	79	19	14	14	14	14	14
20775	43	47	45	48	46	56	132	117	89	88	86	80	20	14	14	14	14	14
21010	43	47	45	48	46	68	129	115	84	87	90	83	22	14	14	14	14	15
21115	28	30	29	30	30	56	81	77	51	56	62	58	15	9	9	9	9	9
21315	31	34	31	32	33	80	81	82	52	58	62	61	17	10	9	9	9	9
21515	31	33	32	31	35	96	82	82	51	57	59	49	18	11	11	11	10	10
21715	37	41	40	43	40	153	119	118	80	75	83	48	21	11	11	12	10	12
21915	36	40	39	42	40	149	117	117	80	83	91	64	22	11	11	12	10	11
22115	36	40	40	41	39	142	116	117	79	92	91	76	24	11	11	12	10	11
22315	36	39	40	41	40	133	117	117	77	93	92	85	25	11	11	12	10	11
22515	35	39	40	40	46	128	119	116	75	93	91	92	28	11	11	12	10	11
22715	35	39	39	39	53	121	116	114	75	93	90	94	33	11	11	11	11	11
22915	35	39	39	38	70	119	118	114	75	93	92	92	40	11	11	11	11	11
23115	34	39	38	38	118	122	118	114	76	94	92	92	52	11	11	11	11	11
23315	33	39	38	38	157	121	120	116	76	94	92	92	71	11	11	11	11	11
23515	32	37	36	36	171	123	123	121	79	96	91	92	90	12	11	11	11	11
23715	22	26	25	25	137	109	111	110	69	82	76	79	78	11	9	9	9	9
23915	22	25	25	27	139	123	131	127	96	110	106	108	108	13	9	9	8	9
24115	22	24	25	41	132	123	131	126	99	110	107	110	109	14	9	9	9	9
24315	22	24	24	103	127	123	131	127	99	111	108	110	110	17	9	9	9	9
24515	21	24	25	144	131	125	128	125	104	112	110	109	109	21	9	9	9	9
24715	20	23	23	112	112	122	116	116	92	109	95	96	91	28	9	9	8	9
24915	24	27	38	133	127	123	125	124	97	107	108	106	108	46	11	11	10	11
25115	23	27	81	132	125	126	123	127	100	108	109	107	107	78	12	11	11	11
25315	23	27	142	129	129	125	127	127	104	107	111	109	108	107	13	11	11	11
25565	29	32	150	133	127	125	129	132	102	105	106	106	106	107	16	12	12	12
25815	32	51	137	131	127	131	127	127	121	124	126	123	122	120	21	12	13	13
26015	42	52	110	105	105	105	105	105	65	66	66	66	66	66	28	17	17	17
26215	42	49	106	105	105	105	105	105	64	64	64	64	64	64	29	16	17	16
26415	42	49	57	57	57	57	57	57	63	63	63	63	63	63	29	16	18	16
26615	41	49	27	27	27	27	27	27	60	61	62	62	62	62	31	16	18	16
26815	41	49	28	27	27	27	27	27	58	61	62	62	62	62	32	16	17	16
27015	41	49	28	27	27	27	27	27	56	61	62	62	62	62	33	17	17	16
27215	40	49	29	27	27	27	27	27	55	61	62	62	62	62	35	17	17	16
27415	39	48	29	27	27	27	27	27	55	61	62	62	62	62	36	17	17	16
27615	39	47	30	27	27	27	27	27	54	61	62	62	62	62	39	17	17	16
27815	38	47	30	27	27	27	27	27	53	61	62	62	62	62	42	17	17	16
28015	38	48	31	27	27	27	27	27	52	61	62	62	62	62	47	17	17	16

## Multiple Lane Drop Freeway Section Geometrics and Capacities



Flow Data for Test Case 3-A : Multiple Lane Drop Congestion Spillback

time period	mainline vol. (vph)
1	1100
2	1100
3	8250
4	8250
5	8250
6	8250
7	8250
8	8250
9	1100
10	1100
11	1100
12	1100
13	1100
14	1100
15	1100
16	1100
17	1100
18	1100



## Model Parameters for Test Case 3-A : Multiple Lane Drop Congestion Spillback

### FRESIM :

- default vehicle set.
- car following factors = 15 to 6.
- acceleration lag = 0.3 sec.
- deceleration lag = 0.3 sec.
- lane change time = 3 sec.
- vehicle generation minimum separation = 0.5 sec.
- percentage of drivers willing to yield = 30 percent.
- minimum non-emergency deceleration = 0.8 ft/sec<sup>2</sup>

### AIMSUN2 :

#### vehicle characteristics:

- length = 4.0 m.
- width = 2.0 m.
- desired speed = 104 kph.
- maximum acceleration = 2.8 m/sec<sup>2</sup>
- maximum deceleration = 4.0 m/sec<sup>2</sup>

#### simulation parameters:

- simulation step size = 0.75sec
- speed acceptance = 0.80
- braking coefficient = 2.0
- zone 1 distance = 100 m
- zone 2 distance = 10 m
- minimum distance between vehicles = 1.5 m
- queue speed = 1.0 m/sec<sup>2</sup>
- queue leaving speed = 4.0 m/sec<sup>2</sup>
- junction visibility = 25 m
- cruising tolerance = 0.80

### FREFLO :

- relaxation time = 75 sec/mile
- anticipation coefficient = 0.25 miles<sup>2</sup>/hour
- capacities as listed on the freeway geometrics
- cubic polynomial speed-density relationship

$$\text{speed} = 81 - 125 (k * 10^{-2}) + 115 (k * 10^{-2})^2 - 49 (k * 10^{-2})^3$$

**FREQ11:**

- capacities as listed on the freeway geometrics
- points on the speed-V/C curve

Points on Upper Portion		of Speed-V/C curve	Points on Lower Portion		of Speed-V/C curve
V/C ratio	speed (mph)		V/C ratio	speed (mph)	
0.00	65		0.00	0	
0.44	65		0.16	6	
0.50	62		0.73	19	
0.73	53		0.86	24	
0.86	46		1.00	34	
1.00	34				

**KRONOS:**

- jam density = 136 vpm
- maximum capacity = 2200 vph
- density at maximum capacity = 68
- density at end of linear region of QK curve = 15 vpm
- flow rate at end of linear region of QK curve = 975 vph
- capacities as listed on the freeway geometrics

**INTEGRATION:**

- mainline freeflow speed = 104 kph
- ramp freeflow speed = 72 kph
- capacities as listed on the freeway geometrics
- speed at capacity for mainline = 54.7 kph
- speed at capacity for ramps = 54.7 kph
- mainline jam density = 84.5 vpkpl
- ramp jam density = 84.5 vpkpl

Flow Data for Test Case 3-B : Multiple Lane Drop Incident Spillback

time period	mainline vol. (vph)
1	4400
2	4400
3	4400
4	4400
5	4400
6	4400
7	4400
8	4400
9	1100
10	1100
11	1100
12	1100
13	1100
14	1100
15	1100
16	1100
17	1100
18	1100

## Model Parameters for Test Case 3-B : Multiple Lane Drop Incident Spillback

### FRESIM :

- default vehicle set.
- car following factors = 15 to 6.
- acceleration lag = 0.3 sec.
- deceleration lag = 0.3 sec.
- lane change time = 3 sec.
- vehicle generation minimum separation = 0.5 sec.
- percentage of drivers willing to yield = 30 percent.
- minimum non-emergency deceleration = 0.8 ft/sec<sup>2</sup>

### AIMSUN2 :

#### vehicle characteristics:

- length = 4.0 m.
- width = 2.0 m.
- desired speed = 104 kph.
- maximum acceleration = 2.8 m/sec<sup>2</sup>
- maximum deceleration = 4.0 m/sec<sup>2</sup>

#### simulation parameters:

- simulation step size = 0.75sec
- speed acceptance = 0.80
- braking coefficient = 2.0
- zone 1 distance = 100 m
- zone 2 distance = 10 m
- minimum distance between vehicles = 1.5 m
- queue speed = 1.0 m/sec<sup>2</sup>
- queue leaving speed = 4.0 m/sec<sup>2</sup>
- junction visibility = 25 m
- cruising tolerance = 0.80

### FREFLO :

- relaxation time = 75 sec/mile
- anticipation coefficient = 0.25 miles<sup>2</sup>/hour
- capacities as listed on the freeway geometrics
- cubic polynomial speed-density relationship

$$\text{speed} = 81 - 125 (k * 10^{-2}) + 115 (k * 10^{-2})^2 - 49 (k * 10^{-2})^3$$

**FREQ11:**

- capacities as listed on the freeway geometrics
- points on the speed-V/C curve

Points on Upper Portion		of Speed-V/C curve	Points on Lower Portion		of Speed-V/C curve
V/C ratio	speed (mph)		V/C ratio	speed (mph)	
0.00	65		0.00	0	
0.44	65		0.16	6	
0.50	62		0.73	19	
0.73	53		0.86	24	
0.86	46		1.00	34	
1.00	34				

**KRONOS:**

- jam density = 136 vpm
- maximum capacity = 2200 vph
- density at maximum capacity = 68
- density at end of linear region of QK curve = 15 vpm
- flow rate at end of linear region of QK curve = 975 vph
- capacities as listed on the freeway geometrics

**INTEGRATION:**

- mainline freeflow speed = 104 kph
- ramp freeflow speed = 72 kph
- capacities as listed on the freeway geometrics
- speed at capacity for mainline = 54.7 kph
- speed at capacity for ramps = 54.7 kph
- mainline jam density = 84.5 vpkpl
- ramp jam density = 84.5 vpkpl

Density Tables for Test Case 3-A : Multiple Lane Drop Congestion Spillback

FRESIM																		
hypothetical test case 3-A																		
DENSITY in vehicles/mile/lane																		
DIST ft	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
200	4	4	28	29	29	30	34	89	94	7	5	4	4	4	4	4	4	4
400	3	3	26	27	28	28	39	104	114	14	4	4	3	3	3	3	3	3
600	3	3	27	28	29	29	53	111	125	34	4	4	3	3	3	3	3	3
800	3	3	26	28	30	31	65	118	133	59	4	3	3	3	3	3	3	3
1000	3	3	26	28	30	33	77	127	135	83	4	3	3	3	3	3	3	4
1200	3	3	26	29	31	39	86	133	138	103	5	3	3	3	3	3	3	3
1400	3	3	25	30	32	45	98	136	137	116	12	3	3	3	3	3	3	3
1600	3	3	25	32	33	51	93	116	119	110	25	3	3	3	3	3	3	3
1800	4	4	30	38	38	64	98	121	123	124	49	4	4	4	4	4	4	4
2000	4	4	29	37	37	67	98	122	121	124	70	4	4	4	4	4	4	4
2200	4	4	28	36	37	74	102	122	123	124	91	4	4	4	4	4	4	4
2400	4	4	27	36	38	82	105	122	123	123	108	6	4	4	4	4	4	4
2600	4	4	27	36	40	89	108	122	123	124	118	12	4	4	4	4	4	4
2800	4	4	26	36	48	94	110	123	123	124	123	25	4	4	4	4	4	4
3000	4	4	25	35	61	93	114	122	124	124	125	44	4	4	4	4	4	4
3200	4	4	24	36	75	91	114	124	122	124	123	65	4	4	4	4	4	4
3400	4	4	24	36	80	89	116	127	125	123	124	84	5	4	4	4	4	4
3600	4	4	23	36	85	90	118	118	125	122	123	104	5	4	4	4	4	4
3800	4	4	23	37	96	91	120	123	125	125	123	117	13	4	4	4	4	4
4000	5	4	22	46	92	94	124	122	125	124	124	124	26	4	4	4	4	4
4200	4	4	22	55	84	97	124	124	123	124	123	127	48	4	4	4	4	4
4400	4	4	21	57	84	99	124	124	124	125	123	126	67	4	4	4	4	4
4600	4	4	22	62	86	103	123	126	125	125	124	127	88	5	4	4	4	4
4800	4	4	21	67	85	104	125	125	125	124	125	127	105	9	4	4	4	4
5000	4	4	20	71	86	107	124	125	125	125	125	126	114	19	4	4	4	4
5200	4	4	20	78	86	111	127	126	126	128	126	129	124	27	4	4	4	4
5400	4	4	20	78	81	106	126	127	127	125	127	129	128	37	4	4	4	4
5600	4	4	21	56	53	88	101	101	103	102	104	103	101	43	4	4	4	4
5800	6	6	23	50	48	94	104	105	105	107	104	107	105	64	6	6	6	6
6000	6	6	21	46	46	97	103	104	105	100	104	105	106	75	5	6	6	5
6200	6	6	19	44	47	100	104	104	105	103	104	105	107	85	6	6	6	6
6400	6	5	18	43	55	102	103	106	104	104	103	105	105	95	8	6	6	6
6600	6	6	17	41	61	103	105	105	105	103	104	105	106	101	11	6	6	6
6800	6	5	16	40	65	103	106	104	105	103	103	105	105	106	15	6	6	6
7000	6	6	15	39	73	104	104	107	105	103	104	105	105	107	24	6	6	6
7200	6	6	15	39	77	104	105	106	104	103	104	104	106	106	36	6	6	6
7400	6	6	14	39	82	103	103	106	105	103	106	104	105	106	46	6	6	5
7600	6	6	14	40	92	102	104	108	104	104	105	104	106	106	57	6	6	6
7800	6	6	13	41	99	103	104	106	105	102	106	104	105	105	67	6	6	6
8000	6	6	12	44	108	103	104	107	105	103	107	104	105	105	77	6	6	5
8200	6	6	12	49	103	103	106	106	106	105	106	104	106	105	88	6	6	5
8400	6	6	11	55	102	103	103	106	105	106	106	105	106	105	97	8	6	6
8600	6	6	11	60	103	103	105	106	107	109	104	104	105	104	102	12	6	6
8800	6	6	10	65	104	103	105	105	105	107	105	104	106	105	105	21	6	6
9000	6	6	10	67	105	104	104	105	105	106	106	105	105	106	106	32	6	6
9200	6	5	9	71	105	105	104	106	105	105	107	105	106	106	106	42	5	6
9400	6	6	9	77	107	106	106	107	109	107	106	106	107	107	107	48	5	6
9600	6	6	8	81	95	95	99	92	87	96	97	99	96	94	91	44	6	6
9800	6	5	8	58	58	56	57	53	52	56	56	59	61	54	55	30	6	5
10000	9	8	11	51	53	52	53	53	53	52	52	53	52	52	52	34	8	9
10200	9	8	11	45	47	47	46	47	48	46	46	46	46	46	46	33	8	9
10400	9	8	10	42	45	44	44	45	46	44	43	44	43	44	43	32	8	9
10600	9	8	10	39	44	41	42	44	44	42	41	43	41	42	40	32	8	9
10800	9	9	10	38	43	40	41	44	44	42	40	42	40	42	40	32	8	9
11000	9	9	10	37	42	40	41	43	43	42	39	41	40	42	39	32	8	8
11200	9	9	9	36	42	39	41	43	42	43	39	41	40	41	40	32	8	9
11400	9	8	9	36	42	40	41	42	42	43	39	41	40	40	40	33	8	9
11600	9	9	9	35	40	41	41	41	41	42	39	40	39	39	40	34	8	8
11800	9	9	8	35	39	39	40	40	40	41	39	38	39	37	39	34	8	8
12000	6	6	6	22	25	26	26	25	26	25	25	25	25	25	25	22	6	6
12200	6	6	5	22	25	25	26	25	25	25	25	25	24	24	25	22	6	6
12400	6	6	6	21	24	24	25	24	25	24	25	24	24	24	24	22	6	6
12600	6	6	6	21	24	24	24	24	25	24	24	24	24	24	24	22	6	6
12800	6	6	6	20	24	24	25	24	25	24	24	24	24	24	24	22	6	6
13000	4	5	4	15	18	18	18	18	19	18	18	18	18	18	17	4	4	4
13200	4	4	4	14	17	18	18	18	18	18	17	17	17	17	17	4	4	4

AIMSUN2																		
hypothetical test case																		
3-A																		
DENSITY in vehicles/mile/lane																		
DIST	TIMESLICE																	
ft	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
200	2	2	18	18	18	21	18	19	2	2	2	2	2	2	2	2	2	2
400	2	2	17	17	18	21	19	19	3	2	2	2	2	2	2	2	2	2
600	2	2	16	17	19	23	24	21	3	2	2	2	2	2	2	2	2	2
800	2	2	16	20	22	28	36	28	4	2	2	2	2	2	2	2	2	2
1000	2	2	17	28	31	39	50	44	8	2	2	2	2	2	2	2	2	2
1200	2	2	20	40	48	55	59	58	15	2	2	2	2	2	2	2	2	2
1400	2	2	29	55	62	63	64	66	25	2	2	2	2	2	2	2	2	2
1600	3	2	33	47	50	49	50	55	31	2	3	3	2	2	3	3	3	3
1800	3	3	18	23	24	25	26	37	32	3	3	3	2	3	3	3	3	3
2000	3	3	16	21	22	23	25	42	46	3	3	3	2	3	3	3	3	3
2200	3	3	15	20	21	23	27	49	62	5	3	3	3	3	3	3	3	3
2400	3	3	15	19	20	24	31	58	80	10	3	3	3	3	3	2	3	3
2600	3	3	14	19	20	26	37	66	95	23	3	3	3	2	3	3	3	3
2800	3	3	14	19	20	28	44	73	100	49	3	3	3	3	3	2	3	3
3000	3	3	13	19	20	30	52	78	103	75	4	3	3	3	3	2	3	3
3200	3	3	13	19	21	34	60	81	106	99	9	3	3	3	3	2	3	3
3400	3	3	13	19	23	38	67	83	109	115	26	3	3	3	3	2	3	2
3600	3	3	13	19	25	45	71	85	111	115	54	3	3	2	3	3	3	2
3800	3	3	12	19	28	53	73	88	112	115	79	4	3	2	3	2	3	2
4000	3	3	12	19	32	60	75	91	113	115	101	9	3	3	3	3	3	2
4200	3	3	12	20	39	66	76	95	114	115	112	27	3	2	3	3	3	2
4400	3	3	11	23	47	69	77	98	114	115	112	54	3	3	3	3	3	2
4600	3	3	11	27	57	71	79	102	114	115	113	78	4	3	3	3	3	2
4800	3	3	11	35	66	74	81	105	115	114	113	100	9	3	3	3	3	2
5000	3	3	12	47	72	74	82	108	115	114	113	112	26	3	3	3	3	2
5200	3	3	13	61	76	76	83	110	115	114	113	113	54	3	3	3	3	2
5400	3	3	20	73	78	78	87	113	116	114	114	113	82	3	3	3	3	3
5600	3	3	22	56	57	58	72	102	104	101	101	102	92	8	3	4	4	3
5800	4	4	12	26	28	30	57	92	93	90	90	91	90	16	4	4	4	4
6000	3	3	11	23	25	27	65	92	93	90	91	91	91	27	3	4	4	3
6200	3	3	10	22	24	28	73	92	93	90	89	92	91	41	3	4	4	3
6400	3	3	9	21	23	31	79	92	93	90	89	90	92	54	3	4	4	3
6600	3	3	9	21	23	36	84	93	92	90	89	91	92	67	4	4	4	3
6800	3	3	9	21	23	43	87	93	92	91	90	91	93	78	6	4	4	3
7000	3	3	9	20	23	52	90	93	91	91	90	91	93	87	10	4	4	3
7200	3	3	8	20	24	61	91	93	91	91	90	91	93	91	19	4	4	3
7400	3	3	8	20	26	69	91	93	91	91	90	91	93	92	32	4	3	3
7600	3	3	8	20	29	76	91	93	90	91	91	91	92	92	44	4	4	3
7800	3	3	8	20	34	83	92	93	91	91	91	92	92	92	57	4	4	4
8000	3	3	7	20	41	86	92	94	90	89	92	92	92	92	70	4	4	4
8200	3	3	7	20	50	89	93	94	90	89	92	94	92	92	80	6	4	4
8400	3	3	7	21	61	90	93	93	90	90	92	93	92	92	87	12	3	3
8600	3	3	6	23	70	90	94	92	90	91	92	92	92	92	91	23	3	3
8800	3	4	6	26	80	92	94	92	91	91	92	93	92	92	91	37	3	4
9000	3	4	6	33	87	92	94	92	92	92	93	93	93	92	92	51	3	4
9200	3	4	6	46	91	93	95	93	92	92	93	94	94	94	93	66	4	3
9400	3	4	6	60	93	94	95	93	94	94	94	95	95	94	94	80	5	3
9600	4	4	7	76	95	95	97	95	95	95	95	97	96	95	95	92	12	4
9800	5	6	9	66	72	71	73	72	73	73	72	73	73	72	73	73	18	6
10000	5	6	7	26	30	30	29	30	30	30	30	30	30	30	30	30	11	6
10200	5	5	6	23	26	25	25	26	26	25	26	25	25	26	25	25	10	6
10400	5	5	6	21	24	24	23	24	24	24	24	23	24	24	24	23	10	6
10600	5	5	6	20	24	23	22	23	23	23	23	22	23	23	23	22	10	5
10800	5	6	5	20	23	23	22	23	23	22	23	22	22	22	22	22	10	5
11000	4	6	5	19	23	23	21	22	22	22	23	21	22	22	22	21	11	5
11200	5	5	5	19	22	22	21	22	22	22	22	21	22	22	22	21	10	5
11400	5	5	5	19	22	22	21	22	22	22	22	21	22	22	22	21	11	5
11600	4	5	5	19	22	22	21	22	22	22	22	21	22	22	22	21	11	5
11800	4	4	4	15	18	18	17	17	17	17	18	17	17	17	17	17	9	4
12000	3	4	3	12	15	15	14	14	15	14	14	14	14	14	14	14	8	4
12200	3	4	3	12	15	14	14	14	14	14	14	14	14	14	14	14	8	4
12400	3	3	3	12	15	14	14	14	15	14	14	14	14	14	14	14	8	4
12600	3	4	3	11	14	14	14	14	14	15	14	14	14	14	14	14	8	4
12800	2	3	3	10	12	12	12	12	12	12	12	12	12	12	12	12	7	3
13000	2	3	3	8	11	11	10	11	11	11	11	11	11	11	11	11	6	3
13200	2	3	3	8	11	11	10	11	11	11	11	10	10	11	11	10	7	3

FREFLO

hypothetical test case  
3-A

DENSITY in vehicles/mile/lane

DIST ft.	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
200	3	3	30	32	32	34	106	126	126	128	51	3	3	3	3	3	3	3
400	3	3	29	32	32	39	102	115	123	114	67	3	3	3	3	3	3	3
600	3	3	28	32	32	48	111	112	118	116	87	4	3	3	3	3	3	3
800	3	3	27	32	32	51	119	113	119	112	105	16	3	3	3	3	3	3
1000	3	3	25	33	33	60	118	113	114	112	112	32	3	3	3	3	3	3
1200	3	3	24	33	33	69	120	117	113	114	119	53	3	3	3	3	3	3
1300	3	3	25	35	35	81	133	119	123	121	125	77	3	3	3	3	3	3
1500	4	4	30	47	47	87	125	111	112	112	116	88	4	4	4	4	4	4
1700	4	4	29	48	48	81	113	101	108	107	108	97	6	4	4	4	4	4
1900	4	4	27	49	49	81	109	102	102	109	101	106	14	4	4	4	4	4
2100	4	4	25	49	50	80	107	104	100	109	98	104	33	4	4	4	4	4
2300	4	4	23	49	50	84	106	108	100	107	101	102	51	4	4	4	4	4
2500	4	4	21	49	52	85	100	109	101	104	104	97	65	4	4	4	4	4
2700	4	4	20	48	58	86	102	106	104	100	106	101	77	4	4	4	4	4
2900	4	4	18	48	60	84	101	102	108	98	106	106	92	4	4	4	4	4
3100	4	4	16	47	59	85	105	98	109	96	106	107	97	13	4	4	4	4
3300	4	4	15	46	61	84	106	97	106	101	107	107	98	33	4	4	4	4
3500	4	4	13	45	60	88	107	100	104	104	106	101	103	43	4	4	4	4
3700	4	4	12	44	60	91	102	105	100	106	104	100	107	61	4	4	4	4
3900	4	4	11	43	61	95	99	106	103	105	103	100	106	76	4	4	4	4
4100	4	4	10	42	61	99	99	106	104	101	103	103	105	91	5	4	4	4
4300	4	4	9	40	65	96	104	100	106	101	104	104	102	106	8	4	4	4
4500	4	4	8	39	69	93	108	97	107	97	105	105	102	104	28	4	4	4
4700	4	4	7	38	72	94	108	99	104	102	106	104	102	100	46	4	4	4
4900	4	4	7	37	78	95	107	103	99	106	106	100	104	97	61	4	4	4
5100	4	4	6	37	87	100	103	104	96	107	107	100	106	100	73	4	4	4
5300	4	4	6	41	97	108	102	108	102	104	105	101	106	108	85	4	4	4
5500	6	6	7	57	107	114	104	113	109	106	108	109	110	115	96	8	6	6
5700	6	6	7	48	72	93	92	97	94	87	94	94	93	98	87	17	6	6
5900	6	6	6	43	64	86	90	91	89	85	87	93	88	90	87	21	6	6
6100	6	6	6	41	60	83	91	88	87	90	82	93	89	87	89	31	6	6
6300	6	6	6	38	58	82	89	90	84	90	84	92	92	83	88	40	6	6
6500	6	6	6	35	57	86	86	89	84	91	86	88	92	83	88	49	6	6
6700	6	6	6	33	56	88	83	89	83	91	89	85	92	86	86	58	6	6
6900	6	6	6	31	55	87	82	90	85	89	89	85	91	90	81	71	6	6
7100	6	6	6	28	54	89	83	89	87	86	89	87	88	91	84	79	6	6
7300	6	6	6	26	54	87	86	85	91	85	87	91	84	91	87	79	6	6
7500	6	6	6	24	56	83	89	85	89	88	84	91	85	89	89	82	12	6
7700	6	6	6	23	57	80	90	87	86	88	83	92	85	85	90	85	15	6
7900	6	6	6	21	57	84	88	88	84	88	85	89	87	84	88	88	25	6
8100	6	6	6	20	56	86	85	90	84	90	88	86	87	85	87	88	33	6
8300	6	6	6	18	55	87	84	89	84	89	89	84	86	88	84	88	43	6
8500	6	6	6	17	55	88	85	88	89	85	91	84	86	89	88	83	52	6
8700	6	6	6	16	53	88	88	85	92	86	87	85	86	88	88	83	64	6
8900	6	6	6	15	55	87	91	82	92	87	84	88	84	84	90	84	70	6
9100	6	6	6	14	58	83	90	86	89	88	85	89	89	83	87	87	76	6
9300	6	6	6	13	61	89	89	92	85	91	86	91	91	86	90	92	84	6
9500	6	6	6	12	69	95	93	95	91	93	95	92	95	94	93	96	91	15
9700	8	9	9	17	85	99	101	98	101	97	102	98	97	101	97	100	101	24
9900	8	9	9	16	82	98	98	98	98	98	98	98	98	98	98	98	98	24
10100	8	9	9	15	86	99	101	99	101	99	101	99	101	99	101	99	101	25
10300	8	9	9	14	87	99	101	99	101	99	101	99	101	99	101	99	101	27
10500	8	9	9	13	88	99	101	99	101	99	101	99	101	99	101	99	101	29
10700	8	9	9	12	89	99	101	99	101	99	101	99	101	99	101	99	101	31
10900	8	9	9	12	89	99	101	99	101	99	101	99	101	99	101	99	101	33
11100	8	9	9	11	90	99	101	99	101	99	101	99	101	99	101	99	101	34
11300	8	9	9	10	91	99	101	99	101	99	101	99	101	99	101	99	101	35
11500	8	9	9	10	91	99	101	99	101	99	101	99	101	99	101	99	101	33
11700	6	6	6	6	21	27	27	27	27	27	27	27	27	27	27	27	27	22
11900	6	6	6	6	21	27	27	27	27	27	27	27	27	27	27	27	27	22
12100	6	6	6	6	20	26	26	26	26	26	26	26	26	26	26	26	26	22
12300	6	6	6	6	19	26	26	26	26	26	26	26	26	26	26	26	26	22
12500	6	6	6	6	18	25	25	25	25	25	25	25	25	25	25	25	25	22
12700	4	4	4	4	13	18	18	18	18	18	18	18	18	18	18	18	18	17
12900	4	4	4	4	12	17	17	17	17	17	17	17	17	17	17	17	17	17
13100	4	4	4	4	12	17	17	17	17	17	17	17	17	17	17	17	17	17



FREQ11

hypothetical test case  
3-A

DENSITY in vehicles/mile/lane

DIST ft	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
200	3	3	18	63	63	63	63	63	7	3	3	3	3	3	3	3	3	3
400	3	3	18	63	63	63	63	63	11	3	3	3	3	3	3	3	3	3
600	3	3	20	63	63	63	63	63	14	3	3	3	3	3	3	3	3	3
800	3	3	21	63	63	63	63	63	16	3	3	3	3	3	3	3	3	3
1000	3	3	22	63	63	63	63	63	17	3	3	3	3	3	3	3	3	3
1200	3	3	23	63	63	63	63	63	17	3	3	3	3	3	3	3	3	3
1400	3	3	24	63	63	63	63	63	18	3	3	3	3	3	3	3	3	3
1600	3	3	26	63	63	63	63	63	73	3	3	3	3	3	3	3	3	3
1800	4	4	32	65	69	69	69	69	79	7	4	4	4	4	4	4	4	4
2000	4	4	32	69	69	69	69	69	79	11	4	4	4	4	4	4	4	4
2200	4	4	34	69	69	69	69	69	79	13	4	4	4	4	4	4	4	4
2400	4	4	34	69	69	69	69	69	79	15	4	4	4	4	4	4	4	4
2600	4	4	34	69	69	69	69	69	79	16	4	4	4	4	4	4	4	4
2800	4	4	37	69	69	69	69	69	79	17	4	4	4	4	4	4	4	4
3000	4	4	37	69	69	69	69	69	79	18	4	4	4	4	4	4	4	4
3200	4	4	37	69	69	69	69	69	79	18	4	4	4	4	4	4	4	4
3400	4	4	39	69	69	69	69	69	79	20	4	4	4	4	4	4	4	4
3600	4	4	39	69	69	69	69	69	79	79	5	4	4	4	4	4	4	4
3800	4	4	39	69	69	69	69	69	79	79	8	4	4	4	4	4	4	4
4000	4	4	39	69	69	69	69	69	79	79	11	4	4	4	4	4	4	4
4200	4	4	42	69	69	69	69	69	79	79	14	4	4	4	4	4	4	4
4400	4	4	42	69	69	69	69	69	79	79	15	4	4	4	4	4	4	4
4600	4	4	42	69	69	69	69	69	79	79	16	4	4	4	4	4	4	4
4800	4	4	46	69	69	69	69	69	79	79	17	4	4	4	4	4	4	4
5000	4	4	46	69	69	69	69	69	79	79	18	4	4	4	4	4	4	4
5200	4	4	46	69	69	69	69	69	79	79	18	4	4	4	4	4	4	4
5400	4	4	50	69	69	69	69	69	79	79	20	4	4	4	4	4	4	4
5600	4	4	50	69	69	69	69	69	79	79	79	5	4	4	4	4	4	4
5800	6	6	56	73	73	73	73	73	81	81	10	6	6	6	6	6	6	6
6000	6	6	56	73	73	73	73	73	81	81	11	6	6	6	6	6	6	6
6200	6	6	56	73	73	73	73	73	81	81	11	6	6	6	6	6	6	6
6400	6	6	56	73	73	73	73	73	81	81	11	6	6	6	6	6	6	6
6600	6	6	56	73	73	73	73	73	81	81	11	6	6	6	6	6	6	6
6800	6	6	56	73	73	73	73	73	81	81	11	6	6	6	6	6	6	6
7000	6	6	56	73	73	73	73	73	81	81	11	6	6	6	6	6	6	6
7200	6	6	61	73	73	73	73	73	81	81	11	6	6	6	6	6	6	6
7400	6	6	61	73	73	73	73	73	81	81	11	6	6	6	6	6	6	6
7600	6	6	61	73	73	73	73	73	81	81	11	6	6	6	6	6	6	6
7800	6	6	61	73	73	73	73	73	81	81	11	6	6	6	6	6	6	6
8000	6	6	61	73	73	73	73	73	81	81	11	6	6	6	6	6	6	6
8200	6	6	61	73	73	73	73	73	81	81	11	6	6	6	6	6	6	6
8400	6	6	61	73	73	73	73	73	81	81	11	6	6	6	6	6	6	6
8600	6	6	67	73	73	73	73	73	81	81	11	6	6	6	6	6	6	6
8800	6	6	67	73	73	73	73	73	81	81	11	6	6	6	6	6	6	6
9000	6	6	67	73	73	73	73	73	81	81	11	6	6	6	6	6	6	6
9200	6	6	67	73	73	73	73	73	81	81	11	6	6	6	6	6	6	6
9400	6	6	67	73	73	73	73	73	81	81	11	6	6	6	6	6	6	6
9600	6	6	67	73	73	73	73	73	81	81	11	6	6	6	6	6	6	6
9800	8	8	73	79	79	79	79	79	65	65	65	65	17	8	8	8	8	8
10000	8	8	73	79	79	79	79	79	65	65	65	65	18	8	8	8	8	8
10200	8	8	79	79	79	79	79	79	65	65	65	65	18	8	8	8	8	8
10400	8	8	79	79	79	79	79	79	65	65	65	65	18	8	8	8	8	8
10600	8	8	32	32	32	32	32	32	65	65	65	65	51	8	8	8	8	8
10800	8	8	18	18	18	18	18	18	65	65	65	65	51	8	8	8	8	8
11000	8	8	18	18	18	18	18	18	65	65	65	65	51	8	8	8	8	8
11200	8	8	18	18	18	18	18	18	65	65	65	65	51	8	8	8	8	8
11400	8	8	18	18	18	18	18	18	65	65	65	65	51	8	8	8	8	8
11600	8	8	18	18	18	18	18	18	65	65	65	65	51	8	8	8	8	8
11800	6	6	11	11	11	11	11	11	27	27	27	27	24	6	6	6	6	6
12000	6	6	11	11	11	11	11	11	27	27	27	27	24	6	6	6	6	6
12200	6	6	11	11	11	11	11	11	27	27	27	27	24	6	6	6	6	6
12400	6	6	11	11	11	11	11	11	27	27	27	27	24	6	6	6	6	6
12600	6	6	11	11	11	11	11	11	27	27	27	27	24	6	6	6	6	6
12800	4	4	8	8	8	8	8	8	18	18	18	18	16	4	4	4	4	4
13000	4	4	8	8	8	8	8	8	18	18	18	18	16	4	4	4	4	4
13200	4	4	8	8	8	8	8	8	18	18	18	18	16	4	4	4	4	4
14000	3	3	7	7	7	7	7	7	14	14	14	14	13	3	3	3	3	3
14500	3	3	8	8	8	8	8	8	14	14	14	14	13	3	3	3	3	3
14800	3	3	7	7	7	7	7	7	12	12	12	12	11	3	3	3	3	3
15300	4	4	9	9	9	9	9	9	15	15	15	15	14	4	4	4	4	4

KRONOS

hypothetical test case  
3-A

DENSITY in vehicles/mile/lane

DIST ft.	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
100	3	3	33	33	33	33	33	33	3	3	3	3	3	3	3	3	3	3
200	3	3	33	33	33	33	33	33	3	3	3	3	3	3	3	3	3	3
300	3	3	33	33	33	33	33	33	3	3	3	3	3	3	3	3	3	3
400	3	3	33	33	33	33	33	33	3	3	3	3	3	3	3	3	3	3
500	3	3	33	33	33	33	33	33	3	3	3	3	3	3	3	3	3	3
600	3	3	33	33	33	33	33	33	3	3	3	3	3	3	3	3	3	3
700	3	3	33	33	33	33	33	33	3	3	3	3	3	3	3	3	3	3
800	3	3	33	33	33	33	33	33	3	3	3	3	3	3	3	3	3	3
900	3	3	33	33	33	33	33	33	3	3	3	3	3	3	3	3	3	3
1000	3	3	33	33	33	33	33	33	3	3	3	3	3	3	3	3	3	3
1100	3	3	33	33	33	33	33	33	3	3	3	3	3	3	3	3	3	3
1200	3	3	33	33	33	33	33	33	3	3	3	3	3	3	3	3	3	3
1300	3	3	33	33	33	33	33	33	3	3	3	3	3	3	3	3	3	3
1400	3	3	33	33	33	33	33	33	3	3	3	3	3	3	3	3	3	3
1500	3	3	33	33	33	33	33	33	3	3	3	3	3	3	3	3	3	3
1600	4	4	40	40	40	35	35	35	4	4	4	4	4	4	4	4	4	4
1700	4	4	50	50	50	66	73	73	4	4	4	4	4	4	4	4	4	4
1800	4	4	50	50	50	90	98	98	4	4	4	4	4	4	4	4	4	4
1900	4	4	50	50	50	106	111	111	4	4	4	4	4	4	4	4	4	4
2000	4	4	50	50	50	113	115	115	4	4	4	4	4	4	4	4	4	4
2100	4	4	50	50	50	115	116	116	4	4	4	4	4	4	4	4	4	4
2200	4	4	50	50	50	116	116	116	4	4	4	4	4	4	4	4	4	4
2300	4	4	50	50	51	116	116	116	4	4	4	4	4	4	4	4	4	4
2400	4	4	50	50	51	116	116	116	4	4	4	4	4	4	4	4	4	4
2500	4	4	50	50	52	116	116	116	4	4	4	4	4	4	4	4	4	4
2600	4	4	49	50	53	116	116	116	10	4	4	4	4	4	4	4	4	4
2700	4	4	49	50	55	116	116	116	34	4	4	4	4	4	4	4	4	4
2800	4	4	49	50	60	116	116	116	73	4	4	4	4	4	4	4	4	4
2900	4	4	49	50	66	116	116	116	103	4	4	4	4	4	4	4	4	4
3000	4	4	48	50	75	116	116	116	113	4	4	4	4	4	4	4	4	4
3100	4	4	48	50	84	116	116	116	116	4	4	4	4	4	4	4	4	4
3200	4	4	47	50	91	116	116	116	116	4	4	4	4	4	4	4	4	4
3300	4	4	47	50	96	116	116	116	116	4	4	4	4	4	4	4	4	4
3400	4	4	47	50	99	116	116	116	116	4	4	4	4	4	4	4	4	4
3500	4	4	46	50	101	116	116	116	116	4	4	4	4	4	4	4	4	4
3600	4	4	46	50	101	116	116	116	116	4	4	4	4	4	4	4	4	4
3700	4	4	45	50	102	116	116	116	116	4	4	4	4	4	4	4	4	4
3800	4	4	45	51	102	116	116	116	116	4	4	4	4	4	4	4	4	4
3900	4	4	44	51	102	116	116	116	116	10	4	4	4	4	4	4	4	4
4000	4	4	44	53	102	116	116	116	116	35	4	4	4	4	4	4	4	4
4100	4	4	43	55	102	116	116	116	116	74	4	4	4	4	4	4	4	4
4200	4	4	42	59	102	116	116	116	116	103	4	4	4	4	4	4	4	4
4300	4	4	42	66	102	116	116	116	116	114	4	4	4	4	4	4	4	4
4400	4	4	41	75	102	116	116	116	116	116	4	4	4	4	4	4	4	4
4500	4	4	41	84	102	116	116	116	116	116	4	4	4	4	4	4	4	4
4600	4	4	40	91	102	116	116	116	116	116	4	4	4	4	4	4	4	4
4700	4	4	40	96	102	116	116	116	116	116	4	4	4	4	4	4	4	4
4800	4	4	39	99	102	116	116	116	116	116	4	4	4	4	4	4	4	4
4900	4	4	38	101	102	116	116	116	116	116	4	4	4	4	4	4	4	4
5000	4	4	38	101	102	116	116	116	116	116	4	4	4	4	4	4	4	4
5100	4	4	37	102	102	116	116	116	116	116	4	4	4	4	4	4	4	4
5200	4	4	37	102	102	116	116	116	116	116	4	4	4	4	4	4	4	4
5300	4	4	36	102	102	116	116	116	116	116	4	4	4	4	4	4	4	4
5400	5	5	48	100	100	120	120	120	120	98	5	5	5	5	5	5	5	5
5500	6	6	56	63	64	107	107	107	107	107	6	6	6	6	6	6	6	6
5600	6	6	54	63	64	107	107	107	107	107	6	6	6	6	6	6	6	6
5700	6	6	53	62	64	107	107	107	107	107	6	6	6	6	6	6	6	6
5800	6	6	51	62	64	107	107	107	107	107	6	6	6	6	6	6	6	6
5900	6	6	49	62	64	107	107	107	107	107	6	6	6	6	6	6	6	6
6000	6	6	48	61	65	107	107	107	107	107	6	6	6	6	6	6	6	6
6100	6	6	46	61	66	107	107	107	107	107	6	6	6	6	6	6	6	6
6200	6	6	45	60	68	107	107	107	107	107	6	6	6	6	6	6	6	6
6300	6	6	43	60	72	107	107	107	107	107	6	6	6	6	6	6	6	6
6400	6	6	42	60	77	107	107	107	107	107	6	6	6	6	6	6	6	6
6500	6	6	41	59	84	107	107	107	107	107	6	6	6	6	6	6	6	6
6600	6	6	40	59	91	107	107	107	107	107	6	6	6	6	6	6	6	6
6700	6	6	39	58	97	107	107	107	107	107	6	6	6	6	6	6	6	6
6800	6	6	38	58	102	107	107	107	107	107	6	6	6	6	6	6	6	6
6900	6	6	36	58	104	107	107	107	107	107	6	6	6	6	6	6	6	6

KRONOS

hypothetical test case  
3-A

DENSITY in vehicles/mile/lane

DIST ft.	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
7000	6	6	36	57	106	107	107	107	107	107	107	7	6	6	6	6	6	6
7100	6	6	35	57	107	107	107	107	107	107	107	18	6	6	6	6	6	6
7200	6	6	34	56	107	107	107	107	107	107	107	45	6	6	6	6	6	6
7300	6	6	33	56	107	107	107	107	107	107	107	78	6	6	6	6	6	6
7400	6	6	32	56	107	107	107	107	107	107	107	98	6	6	6	6	6	6
7500	6	6	31	55	107	107	107	107	107	107	107	105	6	6	6	6	6	6
7600	6	6	30	55	107	107	107	107	107	107	107	107	6	6	6	6	6	6
7700	6	6	29	54	107	107	107	107	107	107	107	107	6	6	6	6	6	6
7800	6	6	29	54	107	107	107	107	107	107	107	107	6	6	6	6	6	6
7900	6	6	28	53	107	107	107	107	107	107	107	107	6	6	6	6	6	6
8000	6	6	27	53	107	107	107	107	107	107	107	107	6	6	6	6	6	6
8100	6	6	27	53	107	107	107	107	107	107	107	107	6	6	6	6	6	6
8200	6	6	26	52	107	107	107	107	107	107	107	107	6	6	6	6	6	6
8300	6	6	25	52	107	107	107	107	107	107	107	107	6	6	6	6	6	6
8400	6	6	25	52	107	107	107	107	107	107	107	107	6	6	6	6	6	6
8500	6	6	24	53	107	107	107	107	107	107	107	107	6	6	6	6	6	6
8600	6	6	23	55	107	107	107	107	107	107	107	107	6	6	6	6	6	6
8700	6	6	23	60	107	107	107	107	107	107	107	107	6	6	6	6	6	6
8800	6	6	22	68	107	107	107	107	107	107	107	107	6	6	6	6	6	6
8900	6	6	22	78	107	107	107	107	107	107	107	107	7	6	6	6	6	6
9000	6	6	21	90	107	107	107	107	107	107	107	107	17	6	6	6	6	6
9100	6	6	20	98	107	107	107	107	107	107	107	107	43	6	6	6	6	6
9200	6	6	20	103	107	107	107	107	107	107	107	107	76	6	6	6	6	6
9300	6	6	19	106	107	107	107	107	107	107	107	107	97	6	6	6	6	6
9400	6	6	19	107	107	107	107	107	107	107	107	107	105	6	6	6	6	6
9500	6	6	19	107	107	107	107	107	107	107	107	107	107	6	6	6	6	6
9600	6	6	21	107	107	107	107	107	107	107	107	107	107	6	6	6	6	6
9700	9	9	31	62	64	65	65	66	66	66	66	66	67	9	9	9	9	9
9800	9	9	30	62	64	65	65	66	66	66	66	66	66	67	9	9	9	9
9900	9	9	29	61	63	64	65	65	66	66	66	66	66	66	9	9	9	9
10000	9	9	28	61	63	64	65	65	66	66	66	66	66	66	9	9	9	9
10100	9	9	28	60	63	64	65	65	66	66	66	66	66	66	9	9	9	9
10200	9	9	27	60	63	64	65	65	66	66	66	66	66	66	9	9	9	9
10300	9	9	26	59	63	64	65	65	66	66	66	66	66	66	9	9	9	9
10400	9	9	25	59	62	64	65	65	66	66	66	66	66	66	9	9	9	9
10500	9	9	24	58	62	64	65	65	66	66	66	66	66	66	9	9	9	9
10600	9	9	23	58	62	64	64	65	65	66	66	66	66	66	9	9	9	9
10700	9	9	22	57	62	63	64	65	65	66	66	66	66	66	9	9	9	9
10800	9	9	21	57	62	63	64	65	65	66	66	66	66	66	9	9	9	9
10900	9	9	20	56	61	63	64	65	65	66	66	66	66	66	9	9	9	9
11000	9	9	19	56	61	63	64	65	65	66	66	66	66	66	9	9	9	9
11100	9	9	18	55	61	63	64	65	65	66	66	66	66	66	9	9	9	9
11200	9	9	17	55	61	63	64	65	65	66	66	66	66	66	9	9	9	9
11300	9	9	16	54	61	63	64	65	65	66	66	66	66	66	9	9	9	9
11400	9	9	15	54	61	63	64	65	65	66	66	66	66	66	9	9	9	9
11500	7	7	13	38	40	41	41	41	41	41	41	41	41	7	7	7	7	7
11600	6	6	11	30	32	32	32	32	32	32	32	32	33	6	6	6	6	6
11700	6	6	9	25	27	27	27	27	27	27	27	27	27	6	6	6	6	6
11800	6	6	7	25	27	27	27	27	27	27	27	27	27	6	6	6	6	6
11900	6	6	6	25	27	27	27	27	27	27	27	27	27	6	6	6	6	6
12000	6	6	6	25	27	27	27	27	27	27	27	27	27	6	6	6	6	6
12100	6	6	6	25	27	27	27	27	27	27	27	27	27	6	6	6	6	6
12200	6	6	6	25	27	27	27	27	27	27	27	27	27	6	6	6	6	6
12300	6	6	6	25	27	27	27	27	27	27	27	27	27	6	6	6	6	6
12400	6	6	6	25	27	27	27	27	27	27	27	27	27	6	6	6	6	6
12500	5	5	5	21	23	23	23	23	23	23	23	23	23	5	5	5	5	5
12600	5	5	5	19	20	20	20	20	20	20	20	20	20	5	5	5	5	5
12700	4	4	4	16	18	18	18	18	18	18	18	18	18	4	4	4	4	4
12800	4	4	4	16	18	18	18	18	18	18	18	18	18	4	4	4	4	4
12900	4	4	4	16	17	18	18	18	18	18	18	18	18	4	4	4	4	4
13000	4	4	4	16	17	18	18	18	18	18	18	18	18	4	4	4	4	4
13100	4	4	4	16	17	18	18	18	18	18	18	18	18	4	4	4	4	4
13200	4	4	4	16	17	18	18	18	18	18	18	18	18	4	4	4	4	4
13300	4	4	4	16	17	18	18	18	18	18	18	18	18	4	4	4	4	4
13400	4	4	4	16	17	18	18	18	18	18	18	18	18	4	4	4	4	4
13500	4	4	4	16	17	18	18	18	18	18	18	18	18	4	4	4	4	4
13600	4	4	4	16	17	18	18	18	18	18	18	18	18	4	4	4	4	4
13700	4	4	4	16	17	18	18	18	18	18	18	18	18	4	4	4	4	4
13800	4	4	4	16	17	18	18	18	18	18	18	18	18	4	4	4	4	4

INTEGRATION

hypothetical test case  
3-A

DENSITY in vehicles/mile/lane

DIST ft	TIMES/LICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
200	3	3	33	33	35	33	152	115	108	9	3	3	3	3	3	3	3	3
400	3	3	30	33	33	31	182	130	128	13	3	3	3	3	3	3	3	3
600	3	3	29	32	32	31	172	130	126	36	3	3	3	3	3	3	3	3
800	3	3	29	32	32	31	156	127	130	129	4	3	3	3	3	3	3	3
1000	3	3	29	32	33	36	144	132	128	129	6	3	3	3	3	3	3	3
1200	3	3	34	42	42	75	138	134	132	136	9	3	3	3	3	3	4	3
1500	3	4	47	68	63	89	117	124	114	121	29	3	4	3	3	4	3	4
1600	4	4	39	50	50	132	115	116	117	118	28	4	4	4	4	5	4	4
1800	4	4	39	51	51	143	119	119	120	120	76	4	4	4	4	4	4	4
2000	4	4	37	50	51	163	120	118	120	119	115	5	4	4	4	5	4	4
2200	4	4	35	49	51	173	118	120	119	122	120	7	4	4	4	4	4	4
2400	4	4	35	49	54	172	118	119	118	120	121	9	4	4	4	4	4	4
2600	4	4	33	48	55	171	120	121	119	120	121	11	4	4	4	4	4	4
2800	4	4	32	48	60	164	119	119	118	119	120	15	4	4	4	4	4	4
3000	4	4	31	48	65	160	119	120	118	120	120	27	4	4	4	4	4	4
3200	4	4	30	48	70	156	119	121	120	119	121	65	4	4	4	4	4	4
3400	4	4	28	47	80	150	119	121	120	120	121	121	5	4	4	4	4	4
3600	5	4	27	48	94	146	118	118	121	118	119	120	7	4	4	4	4	4
3800	5	4	26	49	100	145	118	118	119	117	119	120	9	4	4	4	4	4
4000	5	4	24	53	102	142	121	120	122	120	121	122	11	4	4	4	4	4
4200	5	4	23	60	103	140	118	118	118	122	121	120	15	4	4	4	4	4
4400	4	4	22	73	102	139	120	118	121	118	120	119	27	4	4	4	4	4
4600	5	4	21	85	102	140	119	117	120	119	120	120	68	4	4	4	4	4
4800	4	4	20	96	100	135	120	120	121	119	120	122	118	5	4	4	4	4
5000	4	4	19	103	102	134	122	122	121	121	122	122	121	7	4	5	4	4
5200	5	4	21	106	101	135	126	125	130	128	131	125	127	9	4	5	4	4
5500	5	4	31	105	107	117	109	112	100	107	106	108	105	15	5	5	4	4
5600	6	5	24	63	66	110	106	106	106	105	106	105	106	17	6	6	5	5
5800	6	5	23	61	69	110	109	110	109	109	109	108	109	22	6	6	5	6
6000	6	5	21	59	77	109	109	109	109	108	110	109	109	31	6	6	5	6
6200	6	5	19	57	94	108	109	110	108	108	110	108	110	49	6	6	5	6
6400	6	5	18	56	117	108	109	108	108	108	110	109	108	94	6	6	5	6
6600	6	5	17	55	133	107	109	107	108	108	110	108	108	109	8	6	5	5
6800	6	5	16	54	139	108	109	108	108	108	111	109	108	109	9	6	5	5
7000	6	5	15	54	139	109	109	108	109	108	110	109	108	109	11	6	5	6
7200	6	5	14	53	138	110	109	108	109	108	110	109	108	109	13	6	5	5
7400	6	5	13	52	134	108	107	106	106	106	107	107	106	107	15	6	5	5
7600	6	5	12	50	131	109	109	107	109	107	107	108	108	109	18	6	5	5
7800	6	5	11	49	127	109	109	108	109	109	109	109	109	109	21	6	5	5
8000	6	6	10	48	123	108	110	108	110	108	110	110	110	28	6	6	5	6
8200	6	6	10	46	119	108	110	108	110	109	108	110	110	45	6	6	6	6
8400	6	6	10	46	116	108	110	108	110	109	109	110	109	110	79	6	6	6
8600	6	6	9	48	112	109	110	109	108	110	108	110	109	109	109	7	6	6
8800	6	6	9	62	112	108	110	109	109	110	108	110	109	109	109	9	6	6
9000	6	6	8	92	109	109	110	109	109	110	109	109	109	108	110	10	6	6
9200	6	6	8	109	109	108	110	108	108	110	107	108	107	107	108	12	6	6
9400	6	6	8	109	108	106	105	107	104	106	104	106	106	106	107	14	6	6
9700	6	6	9	108	109	112	112	116	115	115	115	112	112	112	112	22	6	6
9800	9	9	11	60	65	65	65	65	65	65	65	65	65	65	65	25	9	8
10000	9	9	10	58	64	65	65	64	64	64	64	65	65	65	65	27	9	9
10200	9	9	10	54	60	61	61	61	61	61	61	61	61	61	61	28	9	9
10400	9	9	10	51	59	59	59	59	59	59	59	59	59	59	59	30	9	8
10600	9	9	9	50	58	59	59	59	59	59	59	59	59	59	59	31	9	9
10800	9	9	9	49	58	59	59	59	59	59	59	59	59	59	59	32	9	9
11000	9	9	9	46	58	58	59	59	59	59	59	59	59	59	59	34	9	9
11200	9	8	9	45	58	58	58	58	58	58	58	58	58	58	58	37	8	9
11400	9	8	9	42	58	58	58	58	58	58	58	58	58	58	58	39	9	9
11700	9	9	9	39	55	56	56	56	56	56	56	56	56	56	56	45	9	9
11800	6	6	6	22	31	31	31	31	31	31	31	31	31	31	31	25	6	6
12000	6	6	6	21	30	30	30	30	30	30	30	30	30	30	30	27	6	6
12200	6	6	6	20	28	29	29	29	29	29	29	29	29	29	29	27	6	6
12400	5	6	6	19	28	29	28	28	28	28	28	28	28	28	28	28	6	5
12700	6	6	6	19	28	28	28	28	28	28	28	28	28	28	28	28	6	6
12800	4	4	4	13	19	19	19	19	19	19	19	19	19	19	19	19	5	4
13000	4	4	4	13	19	20	20	20	20	20	20	20	20	20	20	20	5	4
13200	4	4	4	13	20	21	21	21	21	21	21	21	21	21	21	21	5	4

Density Tables for Test Case 3-B : Multiple Lane Drop Incident Spillback

FRESIM																	
hypothetical test case																	
3-B																	
DENSITY in vehicles/mile/lane																	
DIST	TIMESLICE																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
200	15	15	15	15	15	15	15	15	4	4	4	4	4	4	4	4	4
400	14	14	14	14	14	14	14	14	4	3	3	3	3	3	3	3	3
600	14	14	14	14	14	14	14	14	4	4	3	3	3	3	3	3	3
800	14	14	14	14	14	14	14	14	4	3	4	3	3	3	3	3	3
1000	14	14	14	14	14	14	14	14	4	3	3	3	3	3	3	3	3
1200	14	14	14	14	14	14	14	14	5	3	3	3	3	3	3	3	3
1400	14	14	14	14	14	14	14	14	5	3	3	3	3	3	3	3	3
1600	14	14	14	14	14	14	14	14	5	3	3	3	3	3	3	3	3
1800	17	17	17	17	17	17	17	17	6	4	4	4	4	4	4	4	4
2000	17	17	18	17	17	17	17	17	7	4	4	4	4	4	4	4	4
2200	17	17	17	17	17	17	17	17	7	4	4	4	4	4	4	4	4
2400	17	17	17	17	17	17	17	17	7	4	4	4	4	4	4	4	4
2600	17	17	17	17	17	17	17	17	7	4	4	4	4	4	4	4	4
2800	17	17	17	17	17	17	17	17	8	4	4	4	4	4	4	4	4
3000	18	17	17	17	17	17	17	17	8	4	4	4	4	4	4	4	4
3200	17	17	17	18	17	18	17	17	8	4	4	4	4	4	4	4	4
3400	18	17	17	18	17	17	17	17	9	4	4	4	4	4	4	4	4
3600	17	17	17	18	17	17	17	17	9	5	4	4	4	4	4	4	4
3800	17	17	17	18	17	18	17	18	26	14	5	4	4	4	4	4	4
4000	18	18	17	18	17	18	17	21	86	64	13	4	4	4	4	4	4
4200	18	18	18	18	18	18	18	30	157	137	41	4	4	4	4	4	4
4400	18	18	18	18	18	18	18	61	213	188	79	4	4	4	4	4	4
4600	19	18	18	19	18	18	19	111	218	183	100	4	4	4	4	4	4
4800	19	18	18	18	18	18	20	149	208	167	110	5	4	4	4	4	4
5000	18	18	18	18	18	18	27	168	190	152	108	9	4	4	4	4	4
5200	18	18	18	18	18	18	57	220	205	170	107	15	4	4	4	4	4
5400	18	18	18	18	18	19	86	208	204	175	101	22	4	4	4	4	4
5600	18	19	18	18	18	19	122	195	170	151	97	30	4	4	4	4	4
5800	24	24	24	24	24	27	193	232	196	174	113	49	6	6	6	6	6
6000	24	24	24	24	24	42	234	230	191	169	112	58	6	5	6	5	6
6200	24	24	23	24	24	72	252	228	185	149	112	68	6	5	6	5	6
6400	24	24	24	23	24	105	227	189	167	147	110	77	6	6	6	5	6
6600	24	23	24	24	24	152	241	197	164	147	108	86	6	6	6	5	6
6800	24	23	24	23	27	190	211	177	152	141	108	95	7	6	6	5	6
7000	24	23	24	23	38	213	200	178	156	140	107	100	10	6	5	6	6
7200	24	23	24	24	65	227	185	152	153	134	108	103	18	6	5	6	6
7400	24	23	24	23	96	210	162	148	152	132	107	104	27	6	6	5	6
7600	24	24	23	24	135	201	161	149	151	131	105	104	39	6	6	6	6
7800	24	24	24	27	169	177	150	154	149	129	105	106	52	6	6	6	6
8000	24	24	23	36	179	163	149	143	151	129	104	105	65	6	6	6	6
8200	24	24	24	64	186	148	147	148	149	122	105	105	74	6	6	5	6
8400	25	25	24	102	166	148	147	149	147	123	104	109	89	6	6	6	6
8600	25	25	26	126	153	147	147	149	119	102	113	98	7	6	6	5	6
8800	27	27	33	134	146	146	150	149	148	115	104	112	99	10	6	6	6
9000	27	27	40	126	147	151	144	151	148	115	104	109	103	17	5	6	6
9200	27	27	50	123	147	153	148	153	147	114	105	107	105	25	6	6	6
9400	28	28	61	126	154	153	151	154	151	113	107	108	107	30	6	5	6
9600	30	32	67	130	149	156	154	154	146	97	91	94	93	29	6	5	6
9800	32	35	63	105	112	116	116	115	105	60	52	53	56	22	6	5	6
10000	45	47	71	110	121	120	118	121	110	55	52	52	52	27	8	8	9
10200	43	44	67	114	127	124	124	129	110	49	47	47	46	26	9	8	9
10400	41	42	76	127	124	128	130	127	106	47	45	45	44	26	8	8	9
10600	40	41	99	103	74	73	81	83	69	44	44	43	42	25	9	8	9
10800	39	40	33	30	27	26	29	28	41	42	44	42	42	26	9	8	9
11000	39	40	26	22	21	21	21	21	32	41	44	41	41	27	9	8	9
11200	40	40	24	19	19	18	19	19	28	41	44	39	41	27	8	8	9
11400	40	40	23	18	18	17	17	17	26	40	43	39	41	28	9	8	9
11600	40	40	23	16	16	17	16	17	24	40	42	39	41	28	8	8	9
11800	40	39	23	16	16	16	17	16	23	39	41	38	39	28	9	8	9
12000	26	25	15	11	10	11	11	11	15	25	26	25	26	18	6	6	6
12200	25	24	15	11	11	10	11	10	15	25	26	25	25	19	6	6	6
12400	25	24	16	10	10	10	10	11	14	24	25	25	24	19	6	6	6
12600	24	23	16	10	11	10	11	10	14	24	25	24	24	19	6	6	6
12800	24	23	16	11	10	11	11	10	14	24	25	24	24	19	6	6	6
13000	18	17	12	8	8	8	8	8	10	18	18	18	18	14	4	4	4
13200	18	17	12	8	8	8	8	8	10	18	18	18	17	14	4	4	4

AIMSUN2																		
hypothetical test case																		
3-B																		
DENSITIES in vehicles/mile/lane																		
DIST ft.	TIMES/LICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
200	9	10	9	10	9	9	10	9	2	2	2	2	1	1	1	2	2	1
400	9	9	9	9	9	8	9	9	2	2	1	1	1	1	1	1	1	1
600	8	9	9	9	9	8	9	9	2	2	1	1	2	1	1	1	1	1
800	8	9	9	9	9	8	9	9	2	2	1	1	1	1	1	1	1	1
1000	8	9	8	9	9	8	9	9	2	2	1	1	2	1	1	1	1	1
1200	8	9	9	9	9	8	9	9	2	2	1	1	2	1	1	1	1	1
1400	10	11	10	11	11	9	11	10	3	2	2	1	2	1	1	2	2	1
1600	14	15	14	16	16	13	17	15	4	2	2	2	2	2	2	2	2	2
1800	11	12	11	12	12	11	12	12	4	2	2	2	2	2	2	2	2	2
2000	11	12	11	12	12	10	11	11	4	2	2	2	2	2	2	2	2	2
2200	10	12	11	11	11	10	11	11	4	2	2	2	2	2	2	2	2	2
2400	10	11	10	11	11	10	11	11	4	2	2	2	2	2	2	2	2	2
2600	10	11	11	11	11	10	11	11	4	2	2	2	2	2	2	2	2	2
2800	10	11	11	11	11	10	11	11	4	2	2	2	2	2	2	2	2	2
3000	10	11	11	11	11	10	11	11	4	2	2	2	2	2	2	2	2	2
3200	10	11	11	11	11	10	11	11	4	2	2	2	2	2	2	2	2	2
3400	10	11	11	11	11	10	11	11	4	2	2	2	2	2	2	2	2	2
3600	10	11	11	11	11	10	11	11	4	2	2	2	2	2	2	2	2	2
3800	10	11	11	11	11	10	11	11	5	2	2	2	2	2	2	2	2	2
4000	10	11	11	11	11	10	11	11	5	2	2	2	2	2	2	2	2	2
4200	10	11	11	11	11	10	10	11	5	2	2	2	2	2	2	2	2	2
4400	10	11	11	11	11	10	10	11	5	2	2	2	2	2	2	2	2	2
4600	10	11	11	11	11	10	10	11	5	2	2	2	2	2	2	2	2	2
4800	10	11	11	11	11	10	10	11	7	2	2	2	2	2	2	2	2	2
5000	10	12	11	11	12	10	11	12	25	16	2	2	2	2	2	2	2	2
5200	11	13	12	13	13	11	12	18	47	35	2	1	2	2	2	2	2	2
5400	15	18	18	19	20	16	16	35	88	75	16	2	2	2	2	2	2	2
5600	24	27	30	30	30	26	27	60	116	98	45	4	3	2	2	3	2	2
5800	15	18	18	18	17	17	20	72	119	118	60	9	3	2	2	2	3	2
6000	14	16	16	16	17	15	24	94	119	117	75	14	3	2	2	2	2	2
6200	13	15	15	15	16	15	32	116	119	117	76	19	3	2	2	2	2	2
6400	13	15	15	15	16	15	49	134	119	117	75	26	3	2	2	2	2	2
6600	13	15	15	14	15	18	75	143	119	116	72	35	3	2	2	2	2	2
6800	13	15	15	14	15	28	108	147	122	117	68	42	3	2	2	2	2	2
7000	12	15	15	14	16	45	129	143	118	115	65	46	7	2	2	2	2	2
7200	12	15	15	14	18	70	141	144	119	113	63	50	12	2	2	2	2	2
7400	12	15	15	14	25	98	143	143	118	110	62	54	16	2	2	2	2	2
7600	12	15	15	15	34	123	144	144	118	106	61	58	19	2	2	2	2	2
7800	12	14	15	17	50	137	143	143	117	102	61	60	25	2	2	2	2	2
8000	12	15	15	20	72	142	143	143	117	97	61	61	34	2	2	2	2	3
8200	12	15	17	26	96	143	144	144	118	93	61	61	42	4	2	2	2	2
8400	13	17	22	34	119	144	145	144	118	88	61	47	8	2	2	2	2	2
8600	14	23	33	52	134	141	142	142	116	84	60	61	52	12	2	2	2	2
8800	17	32	44	79	142	142	143	142	115	81	61	62	56	17	2	2	2	2
9000	21	41	55	105	143	142	143	142	113	79	61	62	59	25	2	2	2	2
9200	29	51	69	127	148	147	148	147	112	80	61	62	61	32	2	2	2	2
9400	46	69	89	134	144	143	144	142	105	80	62	63	63	41	3	2	2	2
9600	67	92	99	137	143	143	143	141	102	81	64	64	64	50	7	3	3	3
9800	58	73	80	128	132	132	132	130	84	61	48	48	48	41	10	3	4	4
10000	23	29	53	121	123	125	124	121	58	25	20	20	20	18	6	4	4	4
10200	20	24	64	123	123	126	124	122	52	21	17	17	17	15	6	3	4	4
10400	18	23	80	124	124	126	124	123	46	19	16	16	16	14	6	3	4	4
10600	18	22	97	127	125	127	126	125	40	18	16	15	15	14	6	3	4	4
10800	17	21	60	73	73	73	73	73	32	18	15	15	15	13	6	3	4	4
11000	17	21	13	13	13	13	13	14	27	18	15	15	15	13	6	3	4	4
11200	17	21	11	10	11	10	11	11	25	18	15	15	15	13	6	3	4	4
11400	16	21	11	10	10	9	10	10	24	18	15	15	15	13	6	3	4	4
11600	16	21	11	9	9	9	9	9	23	18	15	14	15	13	6	3	4	4
11800	13	16	9	7	7	7	7	7	18	14	12	12	12	10	5	3	3	3
12000	11	14	7	6	6	6	6	6	14	12	10	10	10	9	4	2	2	2
12200	11	14	8	6	6	6	6	6	14	12	10	9	10	9	4	2	2	2
12400	11	14	8	6	6	6	6	6	13	12	10	10	10	9	4	2	2	2
12600	11	14	8	6	6	6	6	6	13	12	10	10	10	9	4	2	2	2
12800	9	12	7	5	5	5	5	5	11	10	8	8	8	8	4	2	2	2
13000	8	10	6	4	5	4	4	5	9	9	7	7	7	7	4	2	2	2
13200	8	10	6	4	5	4	4	5	9	9	7	7	7	7	4	2	2	2

FREFLO																		
hypothetical test case																		
3-B																		
DENSITY in vehicles/mile/lane																		
DIST ft.	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
200	13	14	14	14	14	14	13	14	4	3	3	3	3	3	3	3	3	3
400	13	14	14	14	14	14	13	14	4	3	3	3	3	3	3	3	3	3
600	13	14	14	14	14	14	13	14	5	3	3	3	3	3	3	3	3	3
800	13	14	14	14	14	14	13	14	5	3	3	3	3	3	3	3	3	3
1000	13	14	14	14	14	14	13	14	5	3	3	3	3	3	3	3	3	3
1200	13	14	14	14	14	14	13	14	6	3	3	3	3	3	3	3	3	3
1300	14	14	14	14	14	14	14	14	6	3	3	3	3	3	3	3	3	3
1500	17	17	17	17	17	17	17	17	8	4	4	4	4	4	4	4	4	4
1700	17	18	18	17	17	18	17	18	8	4	4	4	4	4	4	4	4	4
1900	17	18	18	17	18	18	17	18	10	4	4	4	4	4	4	4	4	4
2100	18	18	18	18	18	18	18	18	58	8	4	4	4	4	4	4	4	4
2300	18	18	18	18	18	18	18	20	118	41	4	4	4	4	4	4	4	4
2500	18	18	18	18	18	18	18	36	129	77	4	4	4	4	4	4	4	4
2700	18	18	18	18	18	18	18	56	132	95	4	4	4	4	4	4	4	4
2900	18	18	18	18	18	18	18	78	135	100	5	4	4	4	4	4	4	4
3100	18	18	18	18	18	18	18	109	138	104	15	4	4	4	4	4	4	4
3300	18	18	18	18	18	18	28	125	136	113	27	4	4	4	4	4	4	4
3500	18	18	18	18	18	18	47	131	128	112	43	4	4	4	4	4	4	4
3700	18	18	18	18	18	18	72	131	132	113	58	4	4	4	4	4	4	4
3900	18	18	18	18	18	18	97	129	136	107	78	4	4	4	4	4	4	4
4100	18	18	18	18	18	20	119	133	136	108	95	5	4	4	4	4	4	4
4300	18	18	18	18	18	34	130	135	133	105	103	8	4	4	4	4	4	4
4500	18	18	18	18	18	57	135	136	134	110	102	27	4	4	4	4	4	4
4700	18	18	18	18	18	79	135	134	137	107	97	43	4	4	4	4	4	4
4900	18	18	18	18	18	94	127	135	138	107	98	56	4	4	4	4	4	4
5100	18	18	18	18	19	110	126	138	134	102	103	71	4	4	4	4	4	4
5300	18	18	18	18	29	127	129	136	130	105	109	84	4	4	4	4	4	4
5500	25	25	25	25	51	132	134	139	135	110	112	103	10	6	6	6	6	6
5700	26	26	26	26	67	122	121	129	119	92	95	93	12	6	6	6	6	6
5900	26	26	26	26	78	122	117	125	111	87	87	90	20	6	6	6	6	6
6100	26	26	26	26	88	124	117	125	109	90	85	90	30	6	6	6	6	6
6300	26	27	27	26	106	124	117	124	109	88	83	89	38	6	6	6	6	6
6500	26	27	27	27	114	123	120	122	108	84	85	88	50	6	6	6	6	6
6700	27	27	27	28	118	119	122	121	109	83	89	83	57	6	6	6	6	6
6900	27	27	27	38	121	119	119	123	109	83	91	84	67	6	6	6	6	6
7100	27	27	27	52	124	122	118	124	104	84	88	87	74	6	6	6	6	6
7300	27	27	27	69	123	121	120	122	101	88	84	89	77	6	6	6	6	6
7500	27	27	27	77	124	121	121	120	100	91	83	89	85	10	6	6	6	6
7700	27	27	27	88	120	122	120	117	97	90	83	89	86	15	6	6	6	6
7900	27	27	27	104	117	122	118	118	96	89	85	88	89	27	6	6	6	6
8100	27	27	27	113	120	122	119	116	96	85	89	84	89	35	6	6	6	6
8300	27	27	29	119	124	118	121	118	96	83	91	82	89	46	6	6	6	6
8500	27	27	39	124	125	115	122	117	95	84	90	85	85	54	6	6	6	6
8700	27	27	51	122	122	115	118	115	95	88	88	91	85	60	6	6	6	6
8900	27	27	58	121	117	119	116	117	92	90	83	88	88	66	6	6	6	6
9100	27	27	73	122	116	119	115	116	90	89	84	88	90	74	6	6	6	6
9300	27	27	87	118	121	118	116	119	92	91	90	88	92	87	7	6	6	6
9500	30	31	101	118	122	120	120	117	99	92	96	91	94	94	13	6	6	6
9700	51	51	121	125	122	123	122	122	107	98	101	100	97	98	26	9	8	9
9900	54	55	105	110	105	108	113	108	77	68	69	67	68	68	24	9	9	8
10100	56	57	105	109	106	109	111	109	70	60	61	60	60	60	25	8	9	9
10300	57	58	108	105	111	110	109	113	64	57	58	57	57	57	26	9	9	8
10400	58	59	108	103	111	108	115	104	63	56	56	56	56	56	28	8	8	9
10600	58	58	54	55	52	54	53	53	57	56	55	55	55	55	30	9	8	9
10800	57	58	28	26	26	26	26	26	53	55	54	54	54	54	32	8	9	9
11000	56	56	25	21	21	21	21	21	49	54	53	53	53	53	33	9	8	9
11200	53	54	24	19	19	19	19	19	46	51	51	51	51	51	34	9	9	8
11500	47	47	24	18	18	18	18	18	39	46	46	45	45	45	32	9	9	8
11700	29	29	16	11	11	11	11	11	24	28	28	28	28	28	21	6	6	6
11900	28	28	16	11	11	11	11	11	22	27	27	27	27	27	21	6	6	6
12100	27	28	16	11	11	11	11	11	21	27	26	26	26	26	22	6	6	6
12300	27	27	17	11	11	11	11	11	20	26	26	26	26	26	22	6	6	6
12500	26	26	17	11	11	11	11	11	19	25	25	25	25	25	22	6	6	6
12700	18	19	13	8	8	8	8	8	13	18	18	18	18	18	16	4	4	4
12900	18	18	13	8	8	8	8	8	13	18	18	18	18	18	17	4	4	4
13100	18	18	13	8	8	8	8	8	12	17	17	17	17	17	17	4	4	4

FREQ11																		
hypothetical test case																		
3-B																		
DENSITY in vehicles/mile/lane																		
DIST ft	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
200	14	14	14	14	14	14	23	63	7	3	3	3	3	3	3	3	3	3
400	14	14	14	14	14	14	14	29	63	11	3	3	3	3	3	3	3	3
600	14	14	14	14	14	14	14	37	63	14	3	3	3	3	3	3	3	3
800	14	14	14	14	14	14	14	49	63	16	3	3	3	3	3	3	3	3
1000	14	14	14	14	14	14	7	63	63	17	3	3	3	3	3	3	3	3
1200	14	14	14	14	14	10	63	63	17	3	3	3	3	3	3	3	3	3
1400	14	14	14	14	14	14	14	63	63	18	3	3	3	3	3	3	3	3
1600	14	14	14	14	14	14	19	63	63	73	3	3	3	3	3	3	3	3
1800	18	18	18	18	18	28	69	69	79	7	4	4	4	4	4	4	4	4
2000	18	18	18	18	18	32	69	69	79	11	4	4	4	4	4	4	4	4
2200	18	18	18	18	18	39	69	69	79	13	4	4	4	4	4	4	4	4
2400	18	18	18	18	18	46	69	69	79	15	4	4	4	4	4	4	4	4
2600	18	18	18	18	18	55	69	69	79	16	4	4	4	4	4	4	4	4
2800	18	18	18	18	9	69	69	69	79	17	4	4	4	4	4	4	4	4
3000	18	18	18	18	12	69	69	69	79	18	4	4	4	4	4	4	4	4
3200	18	18	18	18	15	69	69	69	79	18	4	4	4	4	4	4	4	4
3400	18	18	18	18	20	69	69	69	79	20	4	4	4	4	4	4	4	4
3600	18	18	18	18	24	69	69	69	79	79	5	4	4	4	4	4	4	4
3800	18	18	18	18	29	69	69	69	79	79	8	4	4	4	4	4	4	4
4000	18	18	18	18	34	69	69	69	79	79	11	4	4	4	4	4	4	4
4200	18	18	18	18	42	69	69	69	79	79	14	4	4	4	4	4	4	4
4400	18	18	18	18	50	69	69	69	79	79	15	4	4	4	4	4	4	4
4600	18	18	18	18	61	69	69	69	79	79	16	4	4	4	4	4	4	4
4800	18	18	18	10	69	69	69	69	79	79	17	4	4	4	4	4	4	4
5000	18	18	18	13	69	69	69	69	79	79	18	4	4	4	4	4	4	4
5200	18	18	18	17	69	69	69	69	79	79	18	4	4	4	4	4	4	4
5400	18	18	18	21	69	69	69	69	79	79	20	4	4	4	4	4	4	4
5600	18	18	18	26	69	69	69	69	79	79	79	5	4	4	4	4	4	4
5800	27	27	27	35	73	73	73	73	81	81	81	10	6	6	6	6	6	6
6000	27	27	27	39	73	73	73	73	81	81	81	13	6	6	6	6	6	6
6200	27	27	27	43	73	73	73	73	81	81	81	14	6	6	6	6	6	6
6400	27	27	27	49	73	73	73	73	81	81	81	16	6	6	6	6	6	6
6600	27	27	27	56	73	73	73	73	81	81	81	17	6	6	6	6	6	6
6800	27	27	27	61	73	73	73	73	81	81	81	17	6	6	6	6	6	6
7000	27	27	27	67	73	73	73	73	81	81	81	18	6	6	6	6	6	6
7200	27	27	14	73	73	73	73	73	81	81	81	19	6	6	6	6	6	6
7400	27	27	17	73	73	73	73	73	81	81	81	19	6	6	6	6	6	6
7600	27	27	19	73	73	73	73	73	81	81	81	19	6	6	6	6	6	6
7800	27	27	22	73	73	73	73	73	81	81	81	20	6	6	6	6	6	6
8000	27	27	25	73	73	73	73	73	81	81	81	21	6	6	6	6	6	6
8200	27	27	28	73	73	73	73	73	81	81	81	21	6	6	6	6	6	6
8400	27	27	32	73	73	73	73	73	81	81	81	21	6	6	6	6	6	6
8600	27	27	35	73	73	73	73	73	81	81	81	21	6	6	6	6	6	6
8800	27	27	41	73	73	73	73	73	81	81	81	21	6	6	6	6	6	6
9000	27	27	46	73	73	73	73	73	81	81	81	21	6	6	6	6	6	6
9200	27	27	49	73	73	73	73	73	81	81	81	21	6	6	6	6	6	6
9400	27	27	56	73	73	73	73	73	81	81	81	21	6	6	6	6	6	6
9600	27	27	61	73	73	73	73	73	81	81	81	21	6	6	6	6	6	6
9800	65	65	73	79	79	79	79	79	65	65	65	65	17	8	8	8	8	8
10000	65	65	73	79	79	79	79	79	65	65	65	65	18	8	8	8	8	8
10200	65	65	79	79	79	79	79	79	65	65	65	65	18	8	8	8	8	8
10400	65	65	79	79	79	79	79	79	65	65	65	65	66	8	8	8	8	8
10600	65	65	32	32	32	32	32	32	65	65	65	65	51	8	8	8	8	8
10800	65	65	18	18	18	18	18	18	65	65	65	65	51	8	8	8	8	8
11000	65	65	18	18	18	18	18	18	65	65	65	65	51	8	8	8	8	8
11200	65	65	18	18	18	18	18	18	65	65	65	65	51	8	8	8	8	8
11400	65	65	18	18	18	18	18	18	65	65	65	65	51	8	8	8	8	8
11600	65	65	18	18	18	18	18	18	65	65	65	65	51	8	8	8	8	8
11800	27	27	11	11	11	11	11	11	27	27	27	27	24	6	6	6	6	6
12000	27	27	11	11	11	11	11	11	27	27	27	27	24	6	6	6	6	6
12200	27	27	11	11	11	11	11	11	27	27	27	27	24	6	6	6	6	6
12400	27	27	11	11	11	11	11	11	27	27	27	27	24	6	6	6	6	6
12600	27	27	11	11	11	11	11	11	27	27	27	27	24	6	6	6	6	6
12800	18	18	8	8	8	8	8	8	18	18	18	18	16	4	4	4	4	4
13000	18	18	8	8	8	8	8	8	18	18	18	18	16	4	4	4	4	4
13200	18	18	8	8	8	8	8	8	18	18	18	18	16	4	4	4	4	4
14000	14	14	7	7	7	7	7	7	14	14	14	14	13	3	3	3	3	3
14500	17	17	8	8	8	8	8	8	14	14	14	14	13	3	3	3	3	3
14800	14	14	7	7	7	7	7	7	12	12	12	12	11	3	3	3	3	3
15300	18	18	9	9	9	9	9	9	15	15	15	15	14	4	4	4	4	4



KRONOS

hypothetical test case  
3-B

DENSITY in vehicles/mile/lane

DIST ft.	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
100	14	14	14	14	14	14	14	14	3	3	3	3	3	3	3	3	3	3
200	14	14	14	14	14	14	14	14	3	3	3	3	3	3	3	3	3	3
300	14	14	14	14	14	14	14	14	3	3	3	3	3	3	3	3	3	3
400	14	14	14	14	14	14	14	14	3	3	3	3	3	3	3	3	3	3
500	14	14	14	14	14	14	14	14	3	3	3	3	3	3	3	3	3	3
600	14	14	14	14	14	14	14	14	3	3	3	3	3	3	3	3	3	3
700	14	14	14	14	14	14	14	14	3	3	3	3	3	3	3	3	3	3
800	14	14	14	14	14	14	14	14	3	3	3	3	3	3	3	3	3	3
900	14	14	14	14	14	14	14	14	3	3	3	3	3	3	3	3	3	3
1000	14	14	14	14	14	14	14	14	3	3	3	3	3	3	3	3	3	3
1100	14	14	14	14	14	14	14	14	3	3	3	3	3	3	3	3	3	3
1200	14	14	14	14	14	14	14	14	3	3	3	3	3	3	3	3	3	3
1300	14	14	14	14	14	14	14	14	3	3	3	3	3	3	3	3	3	3
1400	14	14	14	14	14	14	14	14	3	3	3	3	3	3	3	3	3	3
1500	14	14	14	14	14	14	14	14	3	3	3	3	3	3	3	3	3	3
1600	14	14	14	14	14	14	14	14	4	4	4	4	4	4	4	4	4	4
1700	18	18	18	18	18	18	18	18	4	4	4	4	4	4	4	4	4	4
1800	18	18	18	18	18	18	18	18	4	4	4	4	4	4	4	4	4	4
1900	18	18	18	18	18	18	18	18	4	4	4	4	4	4	4	4	4	4
2000	18	18	18	18	18	18	18	18	4	4	4	4	4	4	4	4	4	4
2100	18	18	18	18	18	18	18	18	4	4	4	4	4	4	4	4	4	4
2200	18	18	18	18	18	18	18	18	4	4	4	4	4	4	4	4	4	4
2300	18	18	18	18	18	18	18	18	4	4	4	4	4	4	4	4	4	4
2400	18	18	18	18	18	18	18	18	4	4	4	4	4	4	4	4	4	4
2500	18	18	18	18	18	18	18	18	4	4	4	4	4	4	4	4	4	4
2600	18	18	18	18	18	18	18	18	4	4	4	4	4	4	4	4	4	4
2700	18	18	18	18	18	18	18	18	4	4	4	4	4	4	4	4	4	4
2800	18	18	18	18	18	18	18	18	19	4	4	4	4	4	4	4	4	4
2900	18	18	18	18	18	18	18	26	5	4	4	4	4	4	4	4	4	4
3000	18	18	18	18	18	18	18	45	19	4	4	4	4	4	4	4	4	4
3100	18	18	18	18	18	18	18	80	57	4	4	4	4	4	4	4	4	4
3200	18	18	18	18	18	18	18	111	100	4	4	4	4	4	4	4	4	4
3300	18	18	18	18	18	18	18	124	122	4	4	4	4	4	4	4	4	4
3400	18	18	18	18	18	18	18	127	126	4	4	4	4	4	4	4	4	4
3500	18	18	18	18	18	18	18	127	127	4	4	4	4	4	4	4	4	4
3600	18	18	18	18	18	18	18	127	127	4	4	4	4	4	4	4	4	4
3700	18	18	18	18	18	18	18	20	127	127	4	4	4	4	4	4	4	4
3800	18	18	18	18	18	18	18	27	127	127	5	4	4	4	4	4	4	4
3900	18	18	18	18	18	18	18	49	127	127	12	4	4	4	4	4	4	4
4000	18	18	18	18	18	18	18	85	127	127	40	4	4	4	4	4	4	4
4100	18	18	18	18	18	18	18	113	127	127	79	4	4	4	4	4	4	4
4200	18	18	18	18	18	18	18	124	127	127	105	4	4	4	4	4	4	4
4300	18	18	18	18	18	18	18	127	127	127	114	4	4	4	4	4	4	4
4400	18	18	18	18	18	18	18	127	127	127	116	4	4	4	4	4	4	4
4500	18	18	18	18	18	18	18	127	127	127	116	4	4	4	4	4	4	4
4600	18	18	18	18	18	18	18	20	127	127	127	116	4	4	4	4	4	4
4700	18	18	18	18	18	18	18	29	127	127	127	116	4	4	4	4	4	4
4800	18	18	18	18	18	18	18	53	127	127	127	116	4	4	4	4	4	4
4900	18	18	18	18	18	18	18	89	127	127	127	116	4	4	4	4	4	4
5000	18	18	18	18	18	18	18	116	127	127	127	116	4	4	4	4	4	4
5100	18	18	18	18	18	18	18	125	127	127	127	116	4	4	4	4	4	4
5200	18	18	18	18	18	18	18	127	127	127	127	116	4	4	4	4	4	4
5300	18	18	18	18	18	18	18	127	127	127	126	116	5	4	4	4	4	4
5400	20	20	20	20	20	134	134	134	132	120	115	5	5	5	5	5	5	5
5500	27	27	27	27	28	124	124	124	122	107	107	6	6	6	6	6	6	6
5600	27	27	27	27	29	124	124	124	121	107	107	6	6	6	6	6	6	6
5700	27	27	27	27	35	124	124	124	121	107	107	6	6	6	6	6	6	6
5800	27	27	27	27	51	124	124	124	120	107	107	6	6	6	6	6	6	6
5900	27	27	27	27	78	124	124	124	120	107	107	6	6	6	6	6	6	6
6000	27	27	27	27	104	124	124	124	119	107	107	6	6	6	6	6	6	6
6100	27	27	27	27	118	124	124	124	118	107	107	6	6	6	6	6	6	6
6200	27	27	27	27	123	124	124	124	118	107	107	6	6	6	6	6	6	6
6300	27	27	27	27	123	124	124	124	117	107	107	6	6	6	6	6	6	6
6400	27	27	27	27	124	124	124	124	117	107	107	6	6	6	6	6	6	6

KRONOS

hypothetical test case  
3-B

DENSITY in vehicles/mile/lane

DIST ft	TIMESLICE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
6500	27	27	27	27	124	124	124	124	116	107	107	6	6	6	6	6	6	6
6600	27	27	27	27	124	124	124	124	115	107	107	6	6	6	6	6	6	6
6700	27	27	27	27	124	124	124	124	115	107	107	6	6	6	6	6	6	6
6800	27	27	27	27	124	124	124	124	114	107	107	6	6	6	6	6	6	6
6900	27	27	27	28	124	124	124	124	113	107	107	6	6	6	6	6	6	6
7000	27	27	27	32	124	124	124	124	113	107	107	8	6	6	6	6	6	6
7100	27	27	27	43	124	124	124	124	112	107	107	22	6	6	6	6	6	6
7200	27	27	27	67	124	124	124	124	112	107	107	52	6	6	6	6	6	6
7300	27	27	27	96	124	124	124	124	111	107	107	83	6	6	6	6	6	6
7400	27	27	27	115	124	124	124	124	111	107	107	100	6	6	6	6	6	6
7500	27	27	27	122	124	124	124	124	111	107	107	106	6	6	6	6	6	6
7600	27	27	27	123	124	124	124	124	110	107	107	107	6	6	6	6	6	6
7700	27	27	27	124	124	124	124	124	110	107	107	107	6	6	6	6	6	6
7800	27	27	27	124	124	124	124	124	110	107	107	107	6	6	6	6	6	6
7900	27	27	27	124	124	124	124	124	109	107	107	107	6	6	6	6	6	6
8000	27	27	27	124	124	124	124	124	109	107	107	107	6	6	6	6	6	6
8100	27	27	27	124	124	124	124	124	109	107	107	107	6	6	6	6	6	6
8200	27	27	28	124	124	124	124	124	109	107	107	107	6	6	6	6	6	6
8300	27	27	30	124	124	124	124	124	108	107	107	107	6	6	6	6	6	6
8400	27	27	38	124	124	124	124	124	108	107	107	107	6	6	6	6	6	6
8500	27	27	57	124	124	124	124	124	108	107	107	107	6	6	6	6	6	6
8600	27	27	85	124	124	124	124	124	108	107	107	107	6	6	6	6	6	6
8700	27	27	109	124	124	124	124	124	108	107	107	107	6	6	6	6	6	6
8800	27	27	120	124	124	124	124	124	108	107	107	107	6	6	6	6	6	6
8900	27	27	123	124	124	124	124	124	108	107	107	107	8	6	6	6	6	6
9000	27	27	123	124	124	124	124	124	108	107	107	107	21	6	6	6	6	6
9100	27	27	124	124	124	124	124	124	108	107	107	107	50	6	6	6	6	6
9200	27	27	124	124	124	124	124	124	108	107	107	107	82	6	6	6	6	6
9300	27	27	124	124	124	124	124	124	108	107	107	107	100	6	6	6	6	6
9400	27	27	124	124	124	124	124	124	108	107	107	107	105	6	6	6	6	6
9500	27	27	124	124	124	124	124	124	108	107	107	107	107	6	6	6	6	6
9600	32	32	131	131	131	131	131	131	108	107	107	107	107	6	6	6	6	6
9700	68	68	116	116	116	116	116	116	73	68	67	67	67	9	9	9	9	9
9800	68	68	116	116	116	116	116	116	73	68	67	67	67	9	9	9	9	9
9900	68	68	116	116	116	116	116	116	72	68	67	67	67	9	9	9	9	9
10000	68	68	116	116	116	116	116	116	72	68	67	67	67	9	9	9	9	9
10100	68	68	116	116	116	116	116	116	70	67	67	67	67	9	9	9	9	9
10200	68	68	116	116	116	116	116	116	70	67	67	67	67	9	9	9	9	9
10300	68	68	116	116	116	116	116	116	68	67	67	67	67	9	9	9	9	9
10400	68	84	116	116	116	116	116	116	68	67	67	67	67	9	9	9	9	9
10500	68	68	68	68	68	68	68	34	66	67	67	67	67	9	9	9	9	9
10600	68	68	68	68	68	68	68	34	66	67	67	67	67	9	9	9	9	9
10700	68	52	18	18	18	18	18	18	64	67	67	67	67	9	9	9	9	9
10800	68	68	18	18	18	18	18	18	64	67	67	67	67	9	9	9	9	9
10900	68	68	18	18	18	18	18	18	62	66	67	67	67	9	9	9	9	9
11000	68	68	18	18	18	18	18	18	62	66	67	67	67	9	9	9	9	9
11100	68	68	18	18	18	18	18	18	61	66	67	67	67	9	9	9	9	9
11200	68	68	18	18	18	18	18	18	61	66	67	67	67	9	9	9	9	9
11300	68	68	18	18	18	18	18	18	59	66	67	67	67	9	9	9	9	9
11400	68	68	18	18	18	18	18	18	59	66	67	67	67	9	9	9	9	9
11500	41	41	15	15	15	15	15	15	40	41	41	41	41	7	7	7	7	7
11600	33	33	13	13	13	13	13	13	32	32	33	33	33	6	6	6	6	6
11700	27	27	11	11	11	11	11	11	26	27	27	27	27	6	6	6	6	6
11800	27	27	11	11	11	11	11	11	26	27	27	27	27	6	6	6	6	6
11900	27	27	11	11	11	11	11	11	26	27	27	27	27	6	6	6	6	6
12000	27	27	11	11	11	11	11	11	26	27	27	27	27	6	6	6	6	6
12100	27	27	11	11	11	11	11	11	26	27	27	27	27	6	6	6	6	6
12200	27	27	11	11	11	11	11	11	26	27	27	27	27	6	6	6	6	6
12300	27	27	11	11	11	11	11	11	26	27	27	27	27	6	6	6	6	6
12400	27	27	11	11	11	11	11	11	25	27	27	27	27	6	6	6	6	6
12500	23	23	10	10	10	10	10	10	22	23	23	23	23	5	5	5	5	5
12600	20	20	9	9	9	9	9	9	19	20	20	20	20	5	5	5	5	5
12700	18	18	9	9	9	9	9	9	17	18	18	18	18	4	4	4	4	4
12800	18	18	9	9	9	9	9	9	16	18	18	18	18	4	4	4	4	4
12900	18	18	9	9	9	9	9	9	16	18	18	18	18	4	4	4	4	4
13000	18	18	9	9	9	9	9	9	16	18	18	18	18	4	4	4	4	4

INTEGRATION																	
hypothetical test case																	
3-B																	
DENSITY in vehicles/mile/lane																	
DIST ft	TIMESLICE																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
200	15	15	15	15	15	16	16	15	4	3	3	3	3	3	3	3	3
400	14	14	15	15	15	15	15	14	4	3	3	3	3	3	3	3	3
600	14	14	15	15	15	15	15	14	4	3	3	3	3	3	3	3	3
800	14	14	14	15	15	15	15	14	4	3	3	3	3	3	3	3	3
1000	15	14	14	15	15	14	15	15	4	3	3	3	3	3	3	3	3
1200	16	15	15	16	16	15	16	16	5	3	3	3	3	4	3	4	3
1500	18	17	17	17	17	16	17	18	5	3	3	3	3	4	3	4	4
1600	19	18	19	19	19	18	19	20	6	4	4	4	4	4	4	4	4
1800	20	19	19	20	19	19	20	20	6	4	4	4	4	4	4	4	4
2000	20	19	19	20	19	19	19	20	6	4	4	4	4	4	4	4	4
2200	20	19	20	20	19	19	19	21	7	4	4	4	4	4	4	4	4
2400	20	19	20	20	20	19	19	21	7	4	4	4	4	4	4	4	4
2600	20	19	20	19	19	19	19	21	7	4	4	4	4	4	4	5	4
2800	19	19	19	19	19	19	19	21	7	4	4	4	4	4	4	5	4
3000	19	19	20	19	19	19	19	21	8	4	4	4	4	4	4	4	4
3200	19	19	19	19	19	19	19	21	8	4	4	4	4	4	4	5	4
3400	19	19	19	19	19	19	19	21	8	4	4	4	4	4	4	5	4
3600	20	19	19	19	19	19	19	22	9	4	4	4	4	4	4	4	4
3800	20	19	19	19	19	19	19	34	9	4	4	4	4	4	4	5	4
4000	20	19	19	19	19	19	19	98	12	4	4	4	4	4	4	5	4
4200	20	19	19	19	19	19	19	137	18	4	4	4	4	4	4	5	4
4400	20	19	19	19	19	19	21	128	40	4	4	4	4	4	4	4	4
4600	20	19	19	19	19	19	46	112	88	4	4	4	4	4	4	4	4
4800	20	19	19	19	19	20	127	110	117	6	4	4	4	4	4	4	4
5000	20	19	19	19	19	25	141	107	122	8	4	4	4	4	4	4	4
5200	22	21	21	22	22	74	131	107	122	10	4	4	4	4	4	4	4
5500	27	27	24	24	25	113	134	109	102	20	5	4	5	4	5	5	4
5600	30	28	27	27	29	138	121	88	104	20	6	6	6	6	6	6	5
5800	31	29	28	28	35	128	123	88	106	29	6	6	6	6	6	6	6
6000	30	28	27	28	63	121	122	87	107	48	6	6	6	6	6	6	6
6200	30	28	27	28	126	122	121	88	106	81	6	6	6	6	6	6	6
6400	29	28	27	28	149	121	120	88	106	107	7	6	6	6	6	6	6
6600	30	28	27	29	139	120	122	87	107	108	9	6	6	6	6	6	6
6800	30	28	27	34	130	123	123	87	107	108	10	6	6	6	6	6	6
7000	29	28	27	65	121	124	122	88	106	108	12	6	6	6	6	6	6
7200	29	28	27	118	122	122	124	89	106	107	14	6	6	6	6	6	6
7400	29	28	27	144	120	125	125	89	107	108	16	6	6	6	6	6	6
7600	29	28	28	138	122	127	126	89	107	108	19	6	6	5	6	6	6
7800	29	28	32	127	120	127	124	90	106	107	24	6	6	5	6	6	6
8000	29	28	64	119	122	125	122	90	106	105	35	6	6	6	6	6	6
8200	29	28	126	120	124	126	122	92	108	106	67	6	6	6	6	6	6
8400	29	28	146	122	125	125	123	93	107	107	107	6	6	5	6	6	6
8600	29	29	136	123	127	125	121	94	109	107	110	8	6	6	6	6	6
8800	29	35	128	123	125	124	123	96	106	107	108	10	6	5	6	6	6
9000	29	75	127	123	126	124	123	97	109	107	110	12	6	6	6	6	6
9200	29	144	130	126	122	127	123	100	108	108	108	13	6	6	6	6	5
9400	34	157	131	129	131	131	129	100	108	107	107	16	6	6	6	6	6
9700	68	152	134	135	123	140	124	121	118	117	115	26	6	6	6	6	6
9800	58	121	105	105	105	105	105	62	65	65	65	27	9	9	9	9	8
10000	58	117	105	105	105	105	105	64	65	65	65	30	9	9	9	9	9
10200	57	113	105	105	105	105	105	65	65	65	65	32	9	9	9	9	8
10400	56	109	105	105	105	105	105	66	65	65	65	33	9	9	9	9	9
10600	56	105	105	105	105	105	105	69	66	65	65	36	9	9	9	9	9
10800	55	57	57	57	57	57	57	65	64	64	64	38	9	9	9	9	9
11000	55	27	27	27	27	27	27	60	61	61	61	41	9	9	9	8	9
11200	55	28	27	27	27	27	27	57	59	59	59	43	9	9	9	9	9
11400	54	29	27	27	27	27	27	56	59	59	59	48	9	9	9	8	9
11700	52	29	26	26	26	26	26	52	56	56	56	53	9	9	9	8	9
11800	30	18	16	16	16	16	16	30	32	32	32	30	6	6	6	5	6
12000	28	19	17	17	17	17	17	28	30	30	30	30	6	6	6	5	6
12200	27	19	17	17	17	17	17	26	29	29	29	29	6	6	6	5	6
12400	27	19	17	17	17	17	17	26	29	28	29	29	7	6	6	5	6
12700	27	19	17	17	17	17	17	25	28	28	28	28	8	6	6	6	6
12800	19	13	11	11	11	11	11	17	20	20	20	19	6	4	4	4	4
13000	19	14	12	12	12	12	12	18	20	20	20	20	6	4	4	4	4
13200	19	14	12	12	12	12	12	18	20	20	20	20	6	4	4	4	4

## Appendix C

### Source Code for Prototype Automatic Simulation System

<code>ikronos.c</code>	C-2
<code>kinit.c</code>	C-9
<code>kronos.c</code>	C-15
<code>main.c</code>	C-29
<code>makefile.c</code>	C-33
<code>translat.c</code>	C-34

# IKRONOS.C

/\* ikronos.c - functions used in initializing the translation structures and allocating memory \*/

#include <string.h> /\* memset() \*/

#include "TDFun.h"

#include "TDIFun.h"

#include "iresult.h"

#include "kronos.h"

ushort

TDKIPPrepareSegment (TDKHeader \* header, TDKSegment \* segment)

```
{
/*****
 * Prepare the structure TDKSegment for later use. This means clearing it
 * and allocating the additional space.
 *****/
*/
ushort nbu;
TDKUnit *unit;
TDKSection *ksection;

/* first clear */
memset ((char *) segment, (char) 0, sizeof (TDKSegment));
/* and now allocate */
for (nbu = 0; nbu < 3; nbu++)
{
    unit = &(segment->units[nbu]);

    ksection = &(unit->main);
    ksection->entrance_flows = (float *)
        calloc ((size_t) header->nbnetstates, sizeof (float));
    if (ksection->entrance_flows == NULL)
        return (MemAlloc);

    ksection = &(unit->entrance);
    ksection->entrance_flows = (float *)
        calloc ((size_t) header->nbnetstates, sizeof (float));
    if (ksection->entrance_flows == NULL)
        return (MemAlloc);

    ksection = &(unit->exit);
    ksection->entrance_flows = (float *)
        calloc ((size_t) header->nbnetstates, sizeof (float));
    if (ksection->entrance_flows == NULL)
        return (MemAlloc);
}

return (0);
}
/***** end TDKIPPrepareSegment *****/
```

void

TDKITTransferFlows (TDKHeader \* header, float \*from, float \*to)

```
{
/*****
 * Copies header->nbnetstates flows from a location to another location.
 *****/
```

```

*/
ushort nb;

for (nb = 0; (int) nb < header->nbnetstates; nb++)
{
    to[nb] = from[nb];
}

return;
}
/*..... end TDKITransferFlows ..... */

```

```

void
TDKIDeallocateSegment (TDKSegment * segment)
{
    /*.....
    * Deallocates the previously allocated structures in TDKSegment.
    *.....
    */
    ushort nbu;
    TDKUnit *unit;
    TDKSection *ksection;

    for (nbu = 0; nbu < 3; nbu++)
    {
        unit = &(segment->units[nbu]);

        ksection = &(unit->main);
        if (ksection->entrance_flows != NULL)
            TDIfree ((void *) ksection->entrance_flows);

        ksection = &(unit->entrance);
        if (ksection->entrance_flows != NULL)
            TDIfree ((void *) ksection->entrance_flows);

        ksection = &(unit->exit);
        if (ksection->entrance_flows != NULL)
            TDIfree ((void *) ksection->entrance_flows);
    }

    return;
}
/*..... end TDKIDeallocateSegment ..... */

```

```

void
TDKIPrintSegment (TDKSegment * segment)
{
    /*.....
    * Prints information inside the TDKSegment structure.
    *.....
    */
    ushort nbu, i;
    TDKUnit *unit;
    TDKSection *ksection;

    printf ("-----\n");
    printf ("Segment type = %d\n", segment->type);
    printf ("Segment index = %d\n", Seg_Num);
    for (nbu = 0; nbu < 3; nbu++)
    {
        unit = &(segment->units[nbu]);

```

```

ksection = &(unit->main);
if ((ksection->entrance_flows != NULL)
    && (ksection->TDsection.idsection > 0))
{
    printf ("Unit # = %d\n", nbu);
    printf ("Section Name = %s\n", ksection->TDsection.name);
    printf ("Main section id = %d\n", ksection->TDsection.idsection);
    printf ("Number of lanes = %d\n", ksection->TDsection.nblanes);
    printf ("Begin distance = %f\n", ksection->dist_begin);
    printf ("End distance = %f\n", ksection->dist_end);
    printf ("Number laterals = %d\n", ksection->nblaterals);
    for (i = 0; i < ksection->nblaterals; i++)
    {
        printf (" Lateral type = %d\n", ksection->laterals[i].type);
        printf (" Lateral width = %f\n", ksection->laterals[i].width);
        printf (" Lateral length = %f\n", ksection->laterals[i].length);
    }
}

ksection = &(unit->entrance);
if ((ksection->entrance_flows != NULL)
    && (ksection->TDsection.idsection > 0))
{
    printf ("Unit # = %d\n", nbu);
    printf ("Entrance section id = %d\n", ksection->TDsection.idsection);
    printf ("Number of lanes = %d\n", ksection->TDsection.nblanes);
    printf ("Begin distance = %f\n", ksection->dist_begin);
    printf ("End distance = %f\n", ksection->dist_end);
    printf ("Number laterals = %d\n", ksection->nblaterals);
    for (i = 0; i < ksection->nblaterals; i++)
    {
        printf (" Lateral type = %d\n", ksection->laterals[i].type);
        printf (" Lateral width = %f\n", ksection->laterals[i].width);
        printf (" Lateral length = %f\n", ksection->laterals[i].length);
    }
}

ksection = &(unit->exit);
if ((ksection->entrance_flows != NULL)
    && (ksection->TDsection.idsection > 0))
{
    printf ("Unit # = %d\n", nbu);
    printf ("Exit section id = %d\n", ksection->TDsection.idsection);
    printf ("Number of lanes = %d\n", ksection->TDsection.nblanes);
    printf ("Begin distance = %f\n", ksection->dist_begin);
    printf ("End distance = %f\n", ksection->dist_end);
    printf ("Number laterals = %d\n", ksection->nblaterals);
    for (i = 0; i < ksection->nblaterals; i++)
    {
        printf (" Lateral type = %d\n", ksection->laterals[i].type);
        printf (" Lateral width = %f\n", ksection->laterals[i].width);
        printf (" Lateral length = %f\n", ksection->laterals[i].length);
    }
}

}

printf ("-----\n");
/*
*/
return;
}
/* :::::::::::::::::::: end TDKIPrintSegment :::::::::::::::::::: */

```

```

void
TDKVClearVSegments (ushort maxvsegments, TDKVSegment * vsegments)
{
/******
 * Clear the vsegments[] array. This is done just for debugging purposes. *
*****
*/
  ushort nbs;
  TDKVSegment *vsegment;

  memset ((char *) vsegments, (char) 0,
          (size_t) maxvsegments * sizeof (TDKVSegment));
/*
  for (nbs = 0; nbs < nbvsegments; nbs++) {
    vsegment = vsegments + nbs;

    vsegment->idsection = 0;
    vsegment->length = vsegment->dist_begin = vsegment->dist_end = 0;
    vsegment->nblanes = 0;
    vsegment->lateral_type = 0;

    vsegment->node_origin_exist = FALSE;
    vsegment->node_origin_entrance_exist = FALSE;
    vsegment->node_origin_entrance_type = 0;
    vsegment->node_origin_entrance_idsection = 0;
    vsegment->node_origin_entrance_nblanes = 0;

    vsegment->node_dest_exist = FALSE;
    vsegment->node_dest_exit_exist = FALSE;
    vsegment->node_dest_exit_type = 0;
    vsegment->node_dest_exit_idsection = 0;
    vsegment->node_dest_exit_nblanes = 0;
  }
*/

  return;
}
/*..... end TDKVClearVSegments .....*/

```

```

ushort
TDKVAllocateFlows (ushort maxvsegments, TDKVSegment * vsegments, ushort nbflows)
{
/******
 * In the vsegments[], allocate enough space for these flows. *
*****
*/
  ushort nbs;
  TDKVSegment *vsegment;

  for (nbs = 0; nbs < maxvsegments; nbs++)
  {
    vsegment = vsegments + nbs;

    if ((vsegment->flows = (float *)
         TDcalloc ((size_t) nbflows, sizeof (float))) == NULL)
      return (MemAlloc);
    if ((vsegment->node_origin_entrance_flows = (float *)
         TDcalloc ((size_t) nbflows, sizeof (float))) == NULL)
      return (MemAlloc);
    if ((vsegment->node_dest_exit_flows = (float *)
         TDcalloc ((size_t) nbflows, sizeof (float))) == NULL)
      return (MemAlloc);
  }
}

```



```

return (0);
}
/*..... end TDKVAllocateFlows ..... */

void
TDKVTransferFlows (ushort idsection, ushort nbnetstates, ushort nbvehmods,
                  ushort nbsections,
                  TDStateSection * ssections, float *flows)
{
/*.....
* Transfers the appropriate flows from ssections[] to flows[].
*.....
*/
  ushort position, nbn, nbe, nbv, nbt;
  float flow;
  TDStateSection *sectionst;
  TDStateEntrance *entrancest;
  TDStateVehMod *vehmodst;

  /* first looks for this section in the array */
  position = TDIsearchStateSection (nbsections, ssections, idsection);
  if (position == 0)
    {
      TDImessage (ERROR, "TDKVTransferFlows: section not found");
    }
  sectionst = ssections + (position - 1);

  for (nbn = 0; nbn < nbnetstates; nbn++)
    {
      flow = 0.;
      for (nbe = 0; nbe < sectionst->nbentrances; nbe++)
        {
          entrancest = &(amp)sectionst->entrances[nbe];

          for (nbv = 0; nbv < nbvehmods; nbv++)
            {
              vehmodst = &(amp)entrancest->vehmods[nbv];

              /* put this value */
              flow += vehmodst->entrance_flows[nbn];
            }
          /* end nbv */
        }
          /* end nbe */

      flows[nbn] = flow;
    }

  return;
}
/*..... end TDKVTransferFlows ..... */

ushort
TDKVShiftLeftVSegments (ushort maxvsegments, ushort * nbvsegments,
                       TDKVSegment * vsegments, ushort nbplaces)
{
/*.....

```

```

* Shifts to the left the contents of the vsegments[] array, in nbplaces. *
* Decreases the nbvsegments accordingly. *
* Errors: *
* ParamError: incorrect parameter *
*****
*/
if (nbplaces < 0 || nbplaces > *nbvsegments)
    return (ParamError);
if (nbplaces == 0)
    return (0);

/* memory copy with overlapping */
memmove ((void *) vsegments, (void *) (vsegments + nbplaces),
        (size_t) (maxvsegments - nbplaces) * sizeof (TDKVSegment));

*nbvsegments -= nbplaces;

/* to ease debugging, clear the last nbplaces */
memset ((void *) (vsegments + (maxvsegments - nbplaces)), (int) 0,
        (size_t) nbplaces * sizeof (TDKVSegment));

return (0);
}
/* ..... end TDKVShiftLeftVSegments ..... */

```

```

void
TDKVPrintVSegments ()
{
    /* .....
    * Displays, to screen, the contents of the vsegments array. *
    * .....
    */
    ushort nbs, nbf;
    float *flows;
    TDKVSegment *vsegment;

    printf ("Number of vsegments: %d\n", (int) nbvsegments);

    for (nbs = 0; nbs < nbvsegments; nbs++)
    {
        vsegment = vsegments + nbs;

        printf ("** VSEGMENT %d Idsection: %d, Length: %.2f (%.1f - %.1f), Nblanes: %d\n",
                (int) nbs + 1, (int) vsegment->idsection, vsegment->length,
                vsegment->dist_begin, vsegment->dist_end,
                (int) vsegment->nblanes);

        flows = vsegment->flows;
        printf ("Flows:");
        for (nbf = 0; nbf < nbnetstates; nbf++)
        {
            printf ("%1f, ", flows[nbf]);
        }
        printf ("\n");

        printf (" lateral: ");
        if (vsegment->lateral_type == 0)
        {
            printf ("NONE\n");
        }
        else if (vsegment->lateral_type == 1)
        {
            printf ("RIGHT ENTRANCE\n");
        }
    }
}

```

```

    }
else if (vsegment->lateral_type == 2)
    {
        printf ("LEFT ENTRANCE\n");
    }
else if (vsegment->lateral_type == 3)
    {
        printf ("RIGHT EXIT\n");
    }
else if (vsegment->lateral_type == 4)
    {
        printf ("LEFT EXIT\n");
    }

if (vsegment->node_origin_exist)
    {
        if (vsegment->node_origin_entrance_exist)
            {
                printf ("Node origin - ENTRANCE");
                if (vsegment->node_origin_entrance_type == 1)
                    {
                        printf ("(right) ");
                    }
                else
                    {
                        printf ("(left) ");
                    }
                printf ("Idsection: %d, Nblanes: %d\n",
                    (int) vsegment->node_origin_entrance_idsection,
                    (int) vsegment->node_origin_entrance_nblanes);

                flows = vsegment->node_origin_entrance_flows;
                printf ("    Flows:");
                for (nbf = 0; nbf < nbnetstates; nbf++)
                    {
                        printf ("%1f, ", flows[nbf]);
                    }
                printf ("\n");
            }
        else
            {
                printf ("Node origin.\n");
            }
    }

if (vsegment->node_dest_exist)
    {
        if (vsegment->node_dest_exit_exist)
            {
                printf ("Node dest - EXIT");
                if (vsegment->node_dest_exit_type == 1)
                    {
                        printf ("(right) ");
                    }
                else
                    {
                        printf ("(left) ");
                    }
                printf ("Idsection: %d, Nblanes: %d\n",
                    (int) vsegment->node_dest_exit_idsection,
                    (int) vsegment->node_dest_exit_nblanes);

                flows = vsegment->node_dest_exit_flows;
                printf ("    Flows:");
                for (nbf = 0; nbf < nbnetstates; nbf++)

```

```

        {
            printf("%.1f, ", flows[nbf]);
        }
        printf("\n");
    }
    else
    {
        printf("Node dest.\n");
    }
}

return;
}
/*..... end TDKVPrintVSegments .....*/

```

## KINIT.C

```

/* kinit.c - functions for initializing KRONOS data structures */

#include "TDFun.h"
#include "TDIFun.h"
#include "kronos.h"

#define DEFAULT_NUM_LANES      3
#define DEFAULT_JAM_DENSITY    186
#define DEFAULT_DENSITY_A     15
#define DEFAULT_FLOW_A        975
#define DEFAULT_CRITICAL_DENSITY 58
#define DEFAULT_JAM_FLOW      0
#define DEFAULT_CAPACITY      2200
#define DEFAULT_PASS_VEH_FRACT 1
#define DEFAULT_LCOMM_VEH_FRACT 0
#define DEFAULT_HCOMM_VEH_FRACT 0
#define DEFAULT_LENGTH        1000
#define DEFAULT_RAMP_CAPACITY  1500
#define DEFAULT_DEMAND         1000
#define DEFAULT_RAMP_DEMAND    100
#define DEFAULT_RAMP_LENGTH    600
#define DEFAULT_METER_LOC      300
#define DEFAULT_LENGTH_BEFORE_WEAVE 200
#define DEFAULT_WEAVE_LENGTH   300
#define DEFAULT_LENGTH_BEFORE_TRANS 400
#define DEFAULT_TRANS_LENGTH   200
#define DEFAULT_METER_RATE     800
#define DEFAULT_DIVERSION_RATE  0
#define DEFAULT_CONGESTED      0
#define DEFAULT_RESTRICTED_CAPACITY 1000
#define DEFAULT_ENTRANCE_PERCENTAGE 100

void
TDKInitGlobals (void)
{
    /*.....
    * Initialize global variables used by KRONOS
    *.....
    */

    DX = 100;
    Total_Sim_Time = 60;
    Num_Time_Slices = 4;
    Time_Slice_Duration = (int)((float) Total_Sim_Time / (float) Num_Time_Slices);
    Max_Delay_Speed = 40;

```

Plotting\_Period = Time\_Slice\_Duration;

```
/* Not used by translator */
DT = 0;
Curr_Time_Slice = 0;
Num_DT_In_Time_Slice = 0;
Curr_DT = 0;
Curr_DT_Idx = 0;
Prev_DT_Idx = 0;
DT_DIV_HR = 0;
DX_DIV_MILE = 0;
MPH = 0;
DX_TIMES_DT_DIV_HR = 0;
DT_TIMES_MILE_DIV_HR = 0;
DX_DIV_MILE_TIMES_DT_DIV_HR = 0;
DT_DIV_DX = 0;
DT_DIV_2DX = 0;
}
/*..... end TDKInitGlobals .....*/
```

void

TDKInitFreeway (FW\_Data\_Struct \* FW\_Ptr)

```
{
/*.....
* Initialize freeway data structure to default values
.....*/
```

```
FW_Ptr->Fraction_Pass_Veh = DEFAULT_PASS_VEH_FRACT;
FW_Ptr->Fraction_LComm_Veh = DEFAULT_LCOMM_VEH_FRACT;
FW_Ptr->Fraction_HComm_Veh = DEFAULT_HCOMM_VEH_FRACT;
FW_Ptr->Jam_Density = DEFAULT_JAM_DENSITY;
FW_Ptr->density_a = DEFAULT_DENSITY_A;
FW_Ptr->flow_a = DEFAULT_FLOW_A;
FW_Ptr->Critical_Density = DEFAULT_CRITICAL_DENSITY;
FW_Ptr->Flow_At_Jam = DEFAULT_JAM_FLOW;
FW_Ptr->Capacity = DEFAULT_CAPACITY;
FW_Ptr->CALC_FUEL_AND_POLLUTION = FALSE;
FW_Ptr->CALC_MOES = FALSE;
FW_Ptr->SAVE_INST_DENSITY = TRUE;
FW_Ptr->SAVE_INST_SPEED = TRUE;
FW_Ptr->SAVE_INST_FLOW = TRUE;
FW_Ptr->SAVE_AVG_DENSITY = TRUE;
FW_Ptr->SAVE_AVG_SPEED = TRUE;
FW_Ptr->SAVE_AVG_FLOW = TRUE;
FW_Ptr->USER_SPECIFIED_EXIT_VOLUME = FALSE;
FW_Ptr->USE_EXIT_CONGESTION = FALSE;
FW_Ptr->USE_ENTRY_CONGESTION = FALSE;
FW_Ptr->ENTRY_CONGESTED = FALSE;
FW_Ptr->Seg = NULL;
FW_Ptr->On_R = NULL;
FW_Ptr->Off_R = NULL;
FW_Ptr->>Weave = NULL;
FW_Ptr->LAD = NULL;
FW_Ptr->Num_Segs = 0;
FW_Ptr->Num_On_Ramps = 0;
FW_Ptr->Num_Off_Ramps = 0;
FW_Ptr->Num_WeavingSegs = 0;
FW_Ptr->Plotting_Period = Plotting_Period;
```

```
/* Not used by translator */
FW_Ptr->Free_Flow_Speed = FW_Ptr->Capacity / FW_Ptr->Critical_Density;
FW_Ptr->EXIT_CONGESTED = FALSE;
FW_Ptr->cd_off = 0;
```

```

FW_Ptr->Num_LADSegs = 0;
FW_Ptr->Num_CDSegs = 0;
FW_Ptr->Num_DXs = 0;
FW_Ptr->Num_Grad_QKs = 0;
FW_Ptr->CD = NULL;
FW_Ptr->Incident = NULL;
/* FW_Ptr->Files = NULL; not a pointer, a structre! */
FW_Ptr->Gradient_QK_Ptr = NULL;
FW_Ptr->Seg_Brdr_Dxs = NULL;
FW_Ptr->Brdr_Qks = NULL;
}
/*..... end TDKInitFreeway ..... */

void
TDKInitSegment (Seg_Struct * Seg_Ptr)
{
/*.....
* Initialize segment data structure to default values
.....
*/

if (Seg_Num == 0)
  Seg_Ptr->Num_Lanes = DEFAULT_NUM_LANES;
else
  Seg_Ptr->Num_Lanes = FW_Ptr->Seg[Seg_Num - 1].Num_Lanes;
Seg_Ptr->Length = DEFAULT_LENGTH;
Seg_Ptr->Type = 1;
Seg_Ptr->Init_Flow = DEFAULT_DEMAND;
Seg_Ptr->Capacity = FW_Ptr->Capacity * Seg_Ptr->Num_Lanes;
Seg_Ptr->INCIDENT = FALSE;
Seg_Ptr->INITIALLY_CONGESTED = FALSE;
strcpy (Seg_Ptr->Name, "");
Seg_Ptr->On_Ramp = NULL;
Seg_Ptr->Off_Ramp = NULL;
Seg_Ptr->>Weaving_Seg = NULL;
Seg_Ptr->LAD_Seg = NULL;
Seg_Ptr->Next_Seg = NULL;
Seg_Ptr->Prev_Seg = NULL;

/* Not used by translator */
Seg_Ptr->Main_K = NULL;
Seg_Ptr->Main_U = NULL;
Seg_Ptr->Adj_K = NULL;
Seg_Ptr->Adj_U = NULL;
Seg_Ptr->Generation = NULL;
Seg_Ptr->Lane_K = NULL;
Seg_Ptr->Lane_U = NULL;
Seg_Ptr->Avg_K = NULL;
Seg_Ptr->Avg_U = NULL;
Seg_Ptr->Avg_Q = NULL;
Seg_Ptr->First_DX = 0;
Seg_Ptr->Last_DX = 0;
Seg_Ptr->Flow_Split_DX = 0;
Seg_Ptr->Flow_Rejoin_DX = 0;
Seg_Ptr->CONNECTED = FALSE;
Seg_Ptr->FIRST_SEG = FALSE;
Seg_Ptr->LAST_SEG = FALSE;
Seg_Ptr->QK = NULL;
Seg_Ptr->CD_Seg = NULL;
Seg_Ptr->MOE = NULL;
Seg_Ptr->Connection = NULL;
}
/*..... end TDKInitSegment ..... */

```

```

void
TDKInitOnRamp (On_Ramp_Struct * OnRamp_Ptr)
{
/*****
* Initialize on ramp data structure to default values
*****/
*/

int i;

OnRamp_Ptr->Length = DEFAULT_RAMP_LENGTH;
OnRamp_Ptr->Meter_Location = DEFAULT_METER_LOC;
OnRamp_Ptr->Demand = DEFAULT_RAMP_DEMAND;
OnRamp_Ptr->METERED = FALSE;
OnRamp_Ptr->Seg = NULL;
OnRamp_Ptr->Capacity = DEFAULT_RAMP_CAPACITY * OnRamp_Ptr->Num_Lanes;

/* allocate memory for time slice arrays */
OnRamp_Ptr->Entrance_Rate_At = (float *) calloc (Num_Time_Slices, sizeof (float));
OnRamp_Ptr->Meter_Rate_At = (float *) calloc (Num_Time_Slices, sizeof (float));

/* set arrays to default values */
for (i = 0; i < Num_Time_Slices; i++)
{
    OnRamp_Ptr->Entrance_Rate_At[i] = DEFAULT_RAMP_DEMAND;
    OnRamp_Ptr->Meter_Rate_At[i] = DEFAULT_METER_RATE;
}

/* Not used by translator */
OnRamp_Ptr->K = NULL;
OnRamp_Ptr->U = NULL;
OnRamp_Ptr->Avg_K = NULL;
OnRamp_Ptr->Avg_Q = NULL;
OnRamp_Ptr->Generation = NULL;
OnRamp_Ptr->Avg_U = NULL;
OnRamp_Ptr->MOE = NULL;
OnRamp_Ptr->Connection = NULL;
/* OnRamp_Ptr->QK = NULL; */
OnRamp_Ptr->Num_Cars_Arrived = 0;
OnRamp_Ptr->Num_Cars_Merged = 0;
OnRamp_Ptr->Max_Q_Size = 0;
OnRamp_Ptr->Max_Q_Length = 0;
OnRamp_Ptr->Avg_Q_Size = 0;
OnRamp_Ptr->Avg_Q_Length = 0;
OnRamp_Ptr->Num_DXs = 0;
OnRamp_Ptr->Num_Accel_DXs = 0;
OnRamp_Ptr->Shared_Two_Lane_Exit_Q = 0;
OnRamp_Ptr->q_rmp = 0;
OnRamp_Ptr->Metering_Rate = 0;
OnRamp_Ptr->CONNECTED = FALSE;
OnRamp_Ptr->mg = FALSE;
}
/***** end TDKInitOnRamp *****/

```

```

void
TDKInitOffRamp (Off_Ramp_Struct * OffRamp_Ptr)
{
/*****
* Initialize on ramp data structure to default values
*****/
*/

```

```

int i;

OffRamp_Ptr->Length = DEFAULT_RAMP_LENGTH;
OffRamp_Ptr->Demand = DEFAULT_RAMP_DEMAND;
OffRamp_Ptr->CONGESTED = FALSE;
OffRamp_Ptr->RESTRICTED_CAPACITY = FALSE;
OffRamp_Ptr->Seg = NULL;
OffRamp_Ptr->Capacity = DEFAULT_RAMP_CAPACITY * OffRamp_Ptr->Num_Lanes;

/* allocate memory for time slice arrays */
OffRamp_Ptr->Exit_Rate_At = (float *) calloc (Num_Time_Slices, sizeof (float));
OffRamp_Ptr->Diversion_Rate_At = (float *) calloc (Num_Time_Slices, sizeof (float));
OffRamp_Ptr->Restricted_Capacity_At = (float *) calloc (Num_Time_Slices, sizeof (float));
OffRamp_Ptr->Through_Demand_At = (float *) calloc (Num_Time_Slices, sizeof (float));
OffRamp_Ptr->Congested_At = (boolean *) calloc (Num_Time_Slices, sizeof (boolean));

/* set arrays to default values */
for (i = 0; i < Num_Time_Slices; i++)
{
    OffRamp_Ptr->Exit_Rate_At[i] = DEFAULT_RAMP_DEMAND;
    OffRamp_Ptr->Diversion_Rate_At[i] = DEFAULT_DIVERSION_RATE;
    OffRamp_Ptr->Restricted_Capacity_At[i] = DEFAULT_RESTRICTED_CAPACITY;
    OffRamp_Ptr->Through_Demand_At[i] = DEFAULT_RAMP_DEMAND * OffRamp_Ptr->Num_Lanes;
    OffRamp_Ptr->Congested_At[i] = DEFAULT_CONGESTED;
}

/* Not used by translator */
OffRamp_Ptr->K = NULL;
OffRamp_Ptr->U = NULL;
OffRamp_Ptr->Avg_K = NULL;
OffRamp_Ptr->Avg_Q = NULL;
OffRamp_Ptr->Generation = NULL;
OffRamp_Ptr->Avg_U = NULL;
OffRamp_Ptr->MOE = NULL;
OffRamp_Ptr->Connection = NULL;
/* OffRamp_Ptr->QK = NULL; */
OffRamp_Ptr->Num_Cars_Departed = 0;
OffRamp_Ptr->Num_Cars_Diverged = 0;
OffRamp_Ptr->Max_Q_Size = 0;
OffRamp_Ptr->Max_Q_Length = 0;
OffRamp_Ptr->Avg_Q_Size = 0;
OffRamp_Ptr->Avg_Q_Length = 0;
OffRamp_Ptr->Meter_Location = 0;
OffRamp_Ptr->Num_DXs = 0;
OffRamp_Ptr->Num_Decel_DXs = 0;
OffRamp_Ptr->Diversion_Rate = 0;
OffRamp_Ptr->qrem = 0;
OffRamp_Ptr->Capacity_Restriction = 0;
OffRamp_Ptr->Flow_Remaining = 0;
OffRamp_Ptr->Through_Demand = 0;
OffRamp_Ptr->CONNECTED = FALSE;
OffRamp_Ptr->METERED = FALSE;
OffRamp_Ptr->dg = FALSE;
}
/*..... end TDKInitOnRamp ..... */

void
TDKInitWeaveSeg (Weaving_Seg_Struct * WeaveSeg_Ptr)
{
/*.....
* Initialize weaving segment data structure to default values
.....*/
}

```



```

*/

int i;

WeaveSeg_Ptr->Length_Before_Weave = DEFAULT_LENGTH_BEFORE_WEAVE;
WeaveSeg_Ptr->Length_Of_Weave = DEFAULT_WEAVE_LENGTH;
WeaveSeg_Ptr->Aux_Lane_Capacity_1 = DEFAULT_CAPACITY;
WeaveSeg_Ptr->Aux_Lane_Capacity_2 = DEFAULT_CAPACITY;
WeaveSeg_Ptr->Aux_Lane_Init_Flow = DEFAULT_RAMP_DEMAND;
WeaveSeg_Ptr->Seg = NULL;

/* allocate memory for time slice arrays */
WeaveSeg_Ptr->Entrance_Percentage_At = (float *) calloc (Num_Time_Slices, sizeof(float));

/* set arrays to default values */
for (i = 0; i < Num_Time_Slices; i++)
{
    WeaveSeg_Ptr->Entrance_Percentage_At[i] = DEFAULT_ENTRANCE_PERCENTAGE;
}

/* Not used by translator */
WeaveSeg_Ptr->Generation = NULL;
WeaveSeg_Ptr->Capacity_Factor = 0;
WeaveSeg_Ptr->Ramp_Flow = 0;
WeaveSeg_Ptr->Diverging_Ratio = 0;
WeaveSeg_Ptr->Num_L1_DXs = 0;
WeaveSeg_Ptr->Num_L2_DXs = 0;
WeaveSeg_Ptr->Aux_K = NULL;
WeaveSeg_Ptr->Aux_U = NULL;
WeaveSeg_Ptr->Aux_Avg_K = NULL;
WeaveSeg_Ptr->Aux_Avg_U = NULL;
WeaveSeg_Ptr->Aux_Avg_Q = NULL;
/* WeaveSeg_Ptr->Adj_QK = NULL; */

}

/*..... end TDKInitWeaveSeg ..... */

void
TDKInitLADSeg (LAD_Seg_Struct * LADSeg_Ptr)
{
/*.....
* Initialize Lane add/drop segment data structure to default values
.....*/
int num_lanes_after_transition;

if ((FW_Ptr->Seg[Seg_Num].Type == RHS_LANE_DROP)
    || (FW_Ptr->Seg[Seg_Num].Type == LHS_LANE_DROP))
    num_lanes_after_transition = FW_Ptr->Seg[Seg_Num].Num_Lanes - 1;
else if ((FW_Ptr->Seg[Seg_Num].Type == RHS_LANE_ADD)
         || (FW_Ptr->Seg[Seg_Num].Type == LHS_LANE_ADD))
    num_lanes_after_transition = FW_Ptr->Seg[Seg_Num].Num_Lanes + 1;

LADSeg_Ptr->Capacity = DEFAULT_CAPACITY * num_lanes_after_transition;
LADSeg_Ptr->Transition_Start = DEFAULT_LENGTH_BEFORE_TRANS;
LADSeg_Ptr->Transition_Length = DEFAULT_TRANS_LENGTH;

/* Not used by translator */
LADSeg_Ptr->First_DX_Of_Transition = 0;
LADSeg_Ptr->Last_DX_Of_Transition = 0;
LADSeg_Ptr->QK = NULL;

}

/*..... end TDKInitLADSeg ..... */

```

## KRONOS.C

```
/* kronos.c - functions for preparing, constructing, and writing to file the KRONOS data structures */
```

```
#include "TDFun.h"
#include "TDIFun.h"
```

```
#define KRONOS_GLOBAL
#include "kronos.h"
```

```
#define FEET(x) x*5280/1608.0
```

```
ushort
```

```
TDKBeginTranslation (TDKHeader * header)
```

```
{
/******
 * Begins a translation session in KRONOS format.
 * 1) Allocate memory for KRONOS structures
 * 2) Calculate global parameters as they are known
*****
*/
```

```
/* Allocate memory for KRONOS freeway structure */
FW_Ptr = (FW_Data_Struct *) calloc (1, sizeof (FW_Data_Struct));
```

```
/* initialize global variables used by KRONOS */
TDKInitGlobals ();
```

```
/* initialize freeway data values to defaults */
TDKInitFreeway (FW_Ptr);
```

```
/* Allocate memory for the different lists */
/* (this will be linked lists) */
FW_Ptr->Seg = (Seg_Struct *) calloc (100, sizeof (Seg_Struct));
FW_Ptr->On_R = (On_Ramp_Struct *) calloc (35, sizeof (On_Ramp_Struct));
FW_Ptr->Off_R = (Off_Ramp_Struct *) calloc (35, sizeof (Off_Ramp_Struct));
FW_Ptr->>Weave = (Weaving_Seg_Struct *) calloc (35, sizeof (Weaving_Seg_Struct));
FW_Ptr->LAD = (LAD_Seg_Struct *) calloc (35, sizeof (LAD_Seg_Struct));
```

```
/* Initialize array indices */
Seg_Num = On_Num = Off_Num = Lad_Num = Weave_Num = 0;
```

```
/* assign global variables as known */
Time_Slice_Duration = header->time_offset;
Num_Time_Slices = header->nbnetstates;
Total_Sim_Time = Num_Time_Slices * Time_Slice_Duration;
Plotting_Period = Time_Slice_Duration;
```

```
return (0);
}
/*..... end TDKBeginTranslation ..... */
```

```
ushort
```

```
TDKCreateSegment (TDKSegment * Gsegment)
```

```
{
/******
 * Given a KRONOS segment description generated by GETRAM, takes care of *
 * really creating it in KRONOS file format. *
*****
*/
```

```
int errcode, i;
```

```

Seg_Struct KSegment;
On_Ramp_Struct KOnRamp;
Off_Ramp_Struct KOffRamp;
Weaving_Seg_Struct KWeaveSeg;
LAD_Seg_Struct KLADSeg;
TDKUnit *unit;
TDKSection *Ksection;
TDSection *Gsection;
float length, width, single_lane_capacity, capacity_factor;

/* Print GETRAM segment parameters (for debugging) */
/*
   TDKPrintSegment(Gsegment);
*/

/* Initialize KRONOS segment structure */
TDKInitSegment (&KSegment);

/* Dereference the main-line GETRAM section */
unit = &(Gsegment->units[0]);
Ksection = &(unit->main);
Gsection = &(Ksection->TDsection);

KSegment.Type = Gsegment->type;
KSegment.Num_Lanes = Gsection->nblanes;
KSegment.Capacity = Gsection->capacity;
if (!strcmp (Gsection->name, ""))
{
    sprintf (Gsection->name, "Segment #%d", Seg_Num + 1);
}
strcpy (KSegment.Name, Gsection->name);
if ((errcode = TDGetSectionDim (Gsection->trapezoid, &length, &width)) != 0)
{
    printf ("Error in TDKCreateSegment calling TDGetSectionDim: %d\n", errcode);
    TDKCommunicateError (errcode);
    return errcode;
}
KSegment.Length = (int) FEET (length);
/* Initial flows ? */

/* Pointer to the previous segment */
if (Seg_Num != 0)
{
    KSegment.Prev_Seg = &(FW_Ptr->Seg[Seg_Num - 1]);
}

FW_Ptr->Seg[Seg_Num] = KSegment;

/* Pointer from the previous segment */
if (Seg_Num != 0)
{
    FW_Ptr->Seg[Seg_Num - 1].Next_Seg = &(FW_Ptr->Seg[Seg_Num]);
}

/* On Ramp information */
switch (FW_Ptr->Seg[Seg_Num].Type)
{
    case PIPELINE:
        break;

    case RHS_SIMPLE_ONRAMP:
    case LHS_SIMPLE_ONRAMP:
    case RHS_1_LANE_ADD_ONRAMP:
    case LHS_1_LANE_ADD_ONRAMP:
    case RHS_2_LANE_ADD_ONRAMP:
    case LHS_2_LANE_ADD_ONRAMP:

```

```

case RHS_WEAVE:
case LHS_WEAVE:
case USER_NEW_WEAVING:

    /* add information to on ramp list */

    /* Determine the number of lanes based on segment type */
    if (FW_Ptr->Seg[Seg_Num].Type == RHS_2_LANE_ADD_ONRAMP
        || FW_Ptr->Seg[Seg_Num].Type == LHS_2_LANE_ADD_ONRAMP)
        KOnRamp.Num_Lanes = 2;
    else
        KOnRamp.Num_Lanes = 1;

    /* initialize on ramp structure */
    TDKInitOnRamp (&KOnRamp);

    /* Dereference the entrance GETRAM section */
    unit = &(Gsegment->units[0]);
    Ksection = &(unit->entrance);
    Gsection = &(Ksection->TDsection);

    KOnRamp.Capacity = Gsection->capacity;
    if ((errcode = TDGetSectionDim (Gsection->trapezoid, &length, &width)) != 0)
    {
        printf ("Error in TDKCreateSegment calling TDGetSectionDim: %d\n", errcode);
        return -1;
    }
    KOnRamp.Length = (int) FEET (length);
    /* Time Slice Demand Information */
    for (i = 0; i < Num_Time_Slices; i++)
    {
        KOnRamp.Entrance_Rate_At[i]
            = Ksection->entrance_flows[i];
    }
    /* Initial Demand (flow) */
    KOnRamp.Demand = Ksection->entrance_flows[0];
    /* METERED ? */
    KOnRamp.METERED = Ksection->metering_exist;
    /* Meter_Location */
    /* I will not attempt to specify a meter location or rates in this version */

    /* Pointer to the corresponding segment */
    KOnRamp.Seg = &(FW_Ptr->Seg[Seg_Num]);

    FW_Ptr->On_R[On_Num] = KOnRamp;

    /* Pointer from the corresponding segment */
    FW_Ptr->Seg[Seg_Num].On_Ramp = &(FW_Ptr->On_R[On_Num]);

    /* Increment on ramp index */
    On_Num++;
    break;

default:
    break;
} /* end on ramp switch */

/* Off Ramp Information */
switch (FW_Ptr->Seg[Seg_Num].Type)
{
case PIPELINE:
    break;

case RHS_SIMPLE_OFFRAMP:
case LHS_SIMPLE_OFFRAMP:
case RHS_1_LANE_DROP_OFFRAMP:

```

```

case LHS_1_LANE_DROP_OFFRAMP:
case RHS_2_LANE_DROP_OFFRAMP:
case LHS_2_LANE_DROP_OFFRAMP:
case RHS_WEAWE:
case LHS_WEAWE:
case USER_NEW_WEAIVING:

    /* add information to off ramp list */

    /* Determine the number of lanes based on segment type */
    if (FW_Ptr->Seg[Seg_Num].Type == RHS_2_LANE_DROP_OFFRAMP
        || FW_Ptr->Seg[Seg_Num].Type == LHS_2_LANE_DROP_OFFRAMP)
        KOffRamp.Num_Lanes = 2;
    else
        KOffRamp.Num_Lanes = 1;

    /* initialize off ramp structure */
    TDKInitOffRamp (&KOffRamp);

    /* Dereference the exit GETRAM section */
    unit = &(Gsegment->units[0]);
    Ksection = &(unit->exit);
    Gsection = &(Ksection->TDsection);

    KOffRamp.Capacity = Gsection->capacity;
    if ((errcode = TDGetSectionDim (Gsection->trapezoid, &length, &width)) != 0)
    {
        printf ("Error in TDKCreateSegment calling TDGetSectionDim: %d\n", errcode);
        return -1;
    }

    KOffRamp.Length = (int) FEET (length);
    /* Time Slice Demand Information */
    for (i = 0; i < Num_Time_Slices; i++)
    {
        KOffRamp.Exit_Rate_At[i]
            = Ksection->entrance_flows[i];
    }

    /* Initial Demand (flow) */
    KOffRamp.Demand = Ksection->entrance_flows[0];
    /* METERED */
    /* Info included for on-ramps, but not for off ramps in KRONOS input file.
       May get added to KRONOS in the future */
    /* OffRamp_Ptr->METERED = Ksection->metering_exist; */
    /* Meter Location */
    /* I will not attempt to specify a meter location or rates in this version */

    /* CONGESTED */
    /* RESTRICTED_CAPACITY */
    /* GETRAM is currently not capable of modeling this information */

    /* Pointer to the corresponding segment */
    KOffRamp.Seg = &(FW_Ptr->Seg[Seg_Num]);

    FW_Ptr->Off_R[Off_Num] = KOffRamp;

    /* Pointer from the corresponding segment */
    FW_Ptr->Seg[Seg_Num].Off_Ramp = &(FW_Ptr->Off_R[Off_Num]);

    /* Increment off ramp index */
    Off_Num++;
    break;

default:
    break;
}
/* end off ramp switch */

```

```

/* Weaving and Lane Add/Drop Information */
switch (FW_Ptr->Seg[Seg_Num].Type)
{
case PIPELINE:
    break;

case RHS_WEAVE:
case LHS_WEAVE:
case USER_NEW_WEAVING:

    /* add information to weaving list */

    /* Subtract a lane from the weaving section because KRONOS
       doesn't count the weaving lane as a "main" lane, and
       reduce the mainline capacity accordingly */
    FW_Ptr->Seg[Seg_Num].Num_Lanes--;
    if (FW_Ptr->Seg[Seg_Num].Type == USER_NEW_WEAVING) /* subtract another lane */
        FW_Ptr->Seg[Seg_Num].Num_Lanes--;
    FW_Ptr->Seg[Seg_Num].Capacity *= (float) FW_Ptr->Seg[Seg_Num].Num_Lanes
        / (float) Gsection->nblanes;

    /* initialize weaving structure */
    TDKInitWeaveSeg (&KWeaveSeg);

    /* Dereference the exit GETRAM section */
    unit = &(Gsegment->units[0]);
    Ksection = &(unit->exit);
    Gsection = &(Ksection->TDsection);

    /* The length before the weaving area is used after the lengths are converted,
       so I will ignore */
    /* KWeaveSeg.Length_Before_Weave = ?; */
    /* The length of the weaving area is simply the length of the segment itself */
    if ((errcode = TDGetSectionDim (Gsection->trapezoid, &length, &width)) != 0)
    {
        printf ("Error in TDKCreateSegment calling TDGetSectionDim: %d\n", errcode);
        return -1;
    }
    KWeaveSeg.Length_Of_Weave = FW_Ptr->Seg[Seg_Num].Length;

    /* Assign the auxillary lane capacity as the GETRAM section capacity
       divided by the number of lanes in the GETRAM section */
    single_lane_capacity = Gsection->capacity / (float) Gsection->nblanes;
    KWeaveSeg.Aux_Lane_Capacity_1 = single_lane_capacity;

    if (FW_Ptr->Seg[Seg_Num].Type == USER_NEW_WEAVING) /* assign auxillary capacity #2 */
        KWeaveSeg.Aux_Lane_Capacity_2 = single_lane_capacity;
    else
        KWeaveSeg.Aux_Lane_Capacity_2 = 0.0;

    /* Set the initial auxillary lane flow to the initial on ramp flow */
    KWeaveSeg.Aux_Lane_Init_Flow = KOnRamp.Entrance_Rate_At[0];

    /* Pointer to the corresponding segment */
    KWeaveSeg.Seg = &(FW_Ptr->Seg[Seg_Num]);

    FW_Ptr->Weave[Weave_Num] = KWeaveSeg;

    /* Pointer from the corresponding segment */
    FW_Ptr->Seg[Seg_Num].Weaving_Seg = &(FW_Ptr->Weave[Weave_Num]);

    /* Increment weave index */
    Weave_Num++;
    break;

case RHS_LANE_DROP:

```

```

case LHS_LANE_DROP:
case RHS_LANE_ADD:
case LHS_LANE_ADD:

    /* add information to lane add/drop list */

    /* add a lane to the KRONOS segment in the case of lane drops.
       because GETRAM counts the number of "main" lanes (not counting the
       laterals, and KRONOS counts the number of lanes at the beginning of
       the segment */
    if ((FW_Ptr->Seg[Seg_Num].Type == RHS_LANE_DROP)
        || (FW_Ptr->Seg[Seg_Num].Type == LHS_LANE_DROP))
        FW_Ptr->Seg[Seg_Num].Num_Lanes++;

    /* initialize weaving structure */
    TDKInitLADSeg (&KLADSeg);

    /* Dereference the exit GETRAM section */
    unit = &(Gsegment->units[0]);
    Ksection = &(unit->main);
    Gsection = &(Ksection->TDsection);

    /* can I assume only one lateral? */
    /* printf("nblaterals = %d\tlength = %f\n",Ksection->nblaterals,
       FEET(Ksection->laterals[0].length)); */
    length = FEET (Ksection->laterals[0].length);
    KLADSeg.Transition_Start = FW_Ptr->Seg[Seg_Num].Length
        - KLADSeg.Transition_Length - (int) length;

    /* Capacity after transition */
    if ((FW_Ptr->Seg[Seg_Num].Type == RHS_LANE_DROP)
        || (FW_Ptr->Seg[Seg_Num].Type == LHS_LANE_DROP))
        capacity_factor = (float) FW_Ptr->Seg[Seg_Num].Num_Lanes - 1
            / (float) FW_Ptr->Seg[Seg_Num].Num_Lanes;
    else if ((FW_Ptr->Seg[Seg_Num].Type == RHS_LANE_ADD)
        || (FW_Ptr->Seg[Seg_Num].Type == LHS_LANE_ADD))
        capacity_factor = (float) FW_Ptr->Seg[Seg_Num].Num_Lanes + 1
            / (float) FW_Ptr->Seg[Seg_Num].Num_Lanes;
    KLADSeg.Capacity = FW_Ptr->Seg[Seg_Num].Capacity * capacity_factor;

    FW_Ptr->LAD[Lad_Num] = KLADSeg;

    /* Pointer from the corresponding segment */
    FW_Ptr->Seg[Seg_Num].LAD_Seg = &(FW_Ptr->LAD[Lad_Num]);

    /* Increment lane add/drop index */
    Lad_Num++;
    break;

case CD_ROAD:
case RHS_SHARED_2_LANE_ONRAMP:
case RHS_SHARED_2_LANE_OFFRAMP:
case LHS_SHARED_2_LANE_ONRAMP:
case LHS_SHARED_2_LANE_OFFRAMP:

    /* report an error, because translator cannot
       handle these types (yet) */
    /*
       errcode = ILLEGAL_SEGMENT_TYPE
       TDKCommunicateError(errcode);
    */

default:
    break;
}

```

/\* end weave and LAD switch \*/

```

/* increment segment index */
Seg_Num++;

return (0);
}
/*..... end TDKCreateSegment .....*/

ushort
TDKPrintFile (TDKHeader * header, FILE * kronos_fp)
{
/*.....
* Print all information to the KRONOS input file.
*.....
*/

int Seg_Index, On_Index, Off_Index, Weave_Index, LAD_Index, Time_Index;
Seg_Struct KSegment;
On_Ramp_Struct KOnRamp;
Off_Ramp_Struct KOffRamp;
Weaving_Seg_Struct KWeaveSeg;
LAD_Seg_Struct KLADSeg;

/* not sure what this is used for, always set to 1 */
int Init_Flow_Rate = 1;

/* version number */
fprintf (kronos_fp, "802\n");

/* network name */
fprintf (kronos_fp, "%s\n", header->network_name);

/* delta x (dx) value */
fprintf (kronos_fp, "%d\n", DX);

/* simulation time */
fprintf (kronos_fp, "%d\n", Total_Sim_Time);

/* time slice duration */
fprintf (kronos_fp, "%d\n", Time_Slice_Duration);

/* initial flow rate, always set to 1 */
fprintf (kronos_fp, "%d\n", Init_Flow_Rate);

/* maximum delay speed */
fprintf (kronos_fp, "%d\n", Max_Delay_Speed);

/* total number of segments */
fprintf (kronos_fp, "%d\n", Seg_Num);

/* density at point "a" */
fprintf (kronos_fp, "%d\n", (int) FW_Ptr->density_a);

/* flow at point "a" */
fprintf (kronos_fp, "%d\n", (int) FW_Ptr->flow_a);

/* critical density */
fprintf (kronos_fp, "%d\n", (int) FW_Ptr->Critical_Density);

/* capacity */
fprintf (kronos_fp, "%d\n", (int) FW_Ptr->Capacity);

/* jam density */
fprintf (kronos_fp, "%d\n", (int) FW_Ptr->Jam_Density);

```



```

/* jam flow */
fprintf(kronos_fp, "%d\n", (int) FW_Ptr->Flow_At_Jam);

/* number of on ramps */
fprintf(kronos_fp, "%d\n", On_Num - Weave_Num);

/* number of off ramps */
fprintf(kronos_fp, "%d\n", Off_Num - Weave_Num);

/* number of weaving segments */
fprintf(kronos_fp, "%d\n", Weave_Num);

/* % of passenger vehicles */
fprintf(kronos_fp, "%d\n", (int) FW_Ptr->Fraction_Pass_Veh * 100);

/* % of light commercial vehicles */
fprintf(kronos_fp, "%d\n", (int) FW_Ptr->Fraction_LComm_Veh * 100);

/* % of heavy commercial vehicles */
fprintf(kronos_fp, "%d\n", (int) FW_Ptr->Fraction_HComm_Veh * 100);

/* segment information */
for (Seg_Index = 0; Seg_Index < Seg_Num; Seg_Index++)
{
    KSegment = FW_Ptr->Seg[Seg_Index];

    /* segment length */
    fprintf(kronos_fp, "%d\n", KSegment.Length);

    /* number of lanes */
    fprintf(kronos_fp, "%d\n", KSegment.Num_Lanes);

    /* segment type */
    fprintf(kronos_fp, "%d\n", KSegment.Type);

    /* capacity */
    fprintf(kronos_fp, "%d\n", (int) KSegment.Capacity);

    /* initial flow */
    fprintf(kronos_fp, "%d\n", (int) KSegment.Init_Flow);

    /* initially congested? */
    if (KSegment.INITIAALLY_CONGESTED)
        fprintf(kronos_fp, "%s\n", "Y");
    else
        fprintf(kronos_fp, "%s\n", "N");

    /* incident? */
    if (KSegment.INCIDENT)
        fprintf(kronos_fp, "%s\n", "Y");
    else
        fprintf(kronos_fp, "%s\n", "N");

    /* segment name */
    fprintf(kronos_fp, "%s\n", KSegment.Name);
}
/* end for segment information */

/* ramp information */
for (Seg_Index = 0; Seg_Index < Seg_Num; Seg_Index++)
{
    KSegment = FW_Ptr->Seg[Seg_Index];

    switch (KSegment.Type)
    {
        case PIPELINE:

```

```

break;

case RHS_SIMPLE_ONRAMP:
case LHS_SIMPLE_ONRAMP:
case RHS_1_LANE_ADD_ONRAMP:
case LHS_1_LANE_ADD_ONRAMP:
  KOnRamp = *(KSegment.On_Ramp);
  fprintf(kronos_fp, "%d\n", 1);
  fprintf(kronos_fp, "%d\n", KOnRamp.Length);
  fprintf(kronos_fp, "%d\n", (int) KOnRamp.Capacity);
  fprintf(kronos_fp, "%d\n", (int) KOnRamp.Demand);
  /* metered? */
  if (KOnRamp.METERED)
    fprintf(kronos_fp, "%s\n", "Y");
  else
    fprintf(kronos_fp, "%s\n", "N");
  fprintf(kronos_fp, "%d\n", KOnRamp.Meter_Location);
  break;

case RHS_2_LANE_ADD_ONRAMP:
case LHS_2_LANE_ADD_ONRAMP:
case RHS_SHARED_2_LANE_ONRAMP:
case LHS_SHARED_2_LANE_ONRAMP:
  KOnRamp = *(KSegment.On_Ramp);
  fprintf(kronos_fp, "%d\n", 2);
  fprintf(kronos_fp, "%d\n", KOnRamp.Length);
  fprintf(kronos_fp, "%d\n", (int) KOnRamp.Capacity);
  fprintf(kronos_fp, "%d\n", (int) KOnRamp.Demand);
  /* metered? */
  if (KOnRamp.METERED)
    fprintf(kronos_fp, "%s\n", "Y");
  else
    fprintf(kronos_fp, "%s\n", "N");
  fprintf(kronos_fp, "%d\n", KOnRamp.Meter_Location);
  break;

case RHS_SIMPLE_OFFRAMP:
case LHS_SIMPLE_OFFRAMP:
case RHS_1_LANE_DROP_OFFRAMP:
case LHS_1_LANE_DROP_OFFRAMP:
  KOffRamp = *(KSegment.Off_Ramp);
  fprintf(kronos_fp, "%d\n", 1);
  fprintf(kronos_fp, "%d\n", KOffRamp.Length);
  fprintf(kronos_fp, "%d\n", (int) KOffRamp.Capacity);
  fprintf(kronos_fp, "%d\n", (int) KOffRamp.Demand);
  /* congested? */
  if (KOffRamp.CONGESTED)
    fprintf(kronos_fp, "%s\n", "Y");
  else
    fprintf(kronos_fp, "%s\n", "N");
  /* capacity restricted? */
  if (KOffRamp.RESTRICTED_CAPACITY)
    fprintf(kronos_fp, "%s\n", "Y");
  else
    fprintf(kronos_fp, "%s\n", "N");
  break;

case RHS_2_LANE_DROP_OFFRAMP:
case LHS_2_LANE_DROP_OFFRAMP:
case RHS_SHARED_2_LANE_OFFRAMP:
case LHS_SHARED_2_LANE_OFFRAMP:
  KOffRamp = *(KSegment.Off_Ramp);
  fprintf(kronos_fp, "%d\n", 2);
  fprintf(kronos_fp, "%d\n", KOffRamp.Length);
  fprintf(kronos_fp, "%d\n", (int) KOffRamp.Capacity);
  fprintf(kronos_fp, "%d\n", (int) KOffRamp.Demand);

```

```

/* congested? */
if (KOffRamp.CONGESTED)
    fprintf(kronos_fp, "%s\n", "Y");
else
    fprintf(kronos_fp, "%s\n", "N");
/* capacity restricted? */
if (KOffRamp.RESTRICTED_CAPACITY)
    fprintf(kronos_fp, "%s\n", "Y");
else
    fprintf(kronos_fp, "%s\n", "N");
break;

case RHS_WEAVE:
case LHS_WEAVE:
    KWeaveSeg = *(KSegment.Weaving_Seg);
    KOnRamp = *(KSegment.On_Ramp);
    KOffRamp = *(KSegment.Off_Ramp);
    fprintf(kronos_fp, "%d\n", KOnRamp.Length);
    fprintf(kronos_fp, "%d\n", KOffRamp.Length);
    fprintf(kronos_fp, "%d\n", (int) KWeaveSeg.Aux_Lane_Capacity_1);
    fprintf(kronos_fp, "%d\n", (int) KWeaveSeg.Aux_Lane_Init_Flow);
    fprintf(kronos_fp, "%d\n", (int) KOnRamp.Capacity);
    fprintf(kronos_fp, "%d\n", (int) KOffRamp.Capacity);
    fprintf(kronos_fp, "%d\n", (int) KOnRamp.Demand);
    fprintf(kronos_fp, "%d\n", (int) KOffRamp.Demand);
    /* metered? */
    if (KOnRamp.METERED)
        fprintf(kronos_fp, "%s\n", "Y");
    else
        fprintf(kronos_fp, "%s\n", "N");
    fprintf(kronos_fp, "%d\n", KOnRamp.Meter_Location);
    /* capacity restricted? */
    if (KOffRamp.RESTRICTED_CAPACITY)
        fprintf(kronos_fp, "%s\n", "Y");
    else
        fprintf(kronos_fp, "%s\n", "N");
    break;

case USER_NEW_WEAVING:
    KWeaveSeg = *(KSegment.Weaving_Seg);
    KOnRamp = *(KSegment.On_Ramp);
    KOffRamp = *(KSegment.Off_Ramp);
    fprintf(kronos_fp, "%d\n", KOnRamp.Length);
    fprintf(kronos_fp, "%d\n", KOffRamp.Length);
    fprintf(kronos_fp, "%d\n", (int) KWeaveSeg.Aux_Lane_Capacity_1);
    fprintf(kronos_fp, "%d\n", (int) KWeaveSeg.Aux_Lane_Init_Flow);
    fprintf(kronos_fp, "%d\n", (int) KOnRamp.Capacity);
    fprintf(kronos_fp, "%d\n", (int) KOffRamp.Capacity);
    fprintf(kronos_fp, "%d\n", (int) KOnRamp.Demand);
    fprintf(kronos_fp, "%d\n", (int) KOffRamp.Demand);
    fprintf(kronos_fp, "%d\n", KWeaveSeg.Length_Before_Weave);
    fprintf(kronos_fp, "%d\n", KWeaveSeg.Length_Of_Weave);
    fprintf(kronos_fp, "%d\n", (int) KWeaveSeg.Aux_Lane_Capacity_2);
    /* metered? */
    if (KOnRamp.METERED)
        fprintf(kronos_fp, "%s\n", "Y");
    else
        fprintf(kronos_fp, "%s\n", "N");
    fprintf(kronos_fp, "%d\n", KOnRamp.Meter_Location);
    /* capacity restricted? */
    if (KOffRamp.RESTRICTED_CAPACITY)
        fprintf(kronos_fp, "%s\n", "Y");
    else
        fprintf(kronos_fp, "%s\n", "N");
    break;

```

```

    case RHS_LANE_DROP:
    case LHS_LANE_DROP:
    case RHS_LANE_ADD:
    case LHS_LANE_ADD:
        KLADSeg = *(KSegment.LAD_Seg);
        fprintf(kronos_fp, "%d\n", KLADSeg.Transition_Start);
        fprintf(kronos_fp, "%d\n", KLADSeg.Transition_Length);
        fprintf(kronos_fp, "%d\n", (int) KLADSeg.Capacity);
        break;

    default:
        break;

} /* switch */

} /* end for ramp information */

/* print global variables related to demand */

/* use downstream demand? */
if (FW_Ptr->USER_SPECIFIED_EXIT_VOLUME)
    fprintf(kronos_fp, "%s\n", "Y");
else
    fprintf(kronos_fp, "%s\n", "N");

/* use upstream congestion? */
if (FW_Ptr->USE_ENTRY_CONGESTION)
    fprintf(kronos_fp, "%s\n", "Y");
else
    fprintf(kronos_fp, "%s\n", "N");

/* use downstream congestion? */
if (FW_Ptr->USE_EXIT_CONGESTION)
    fprintf(kronos_fp, "%s\n", "Y");
else
    fprintf(kronos_fp, "%s\n", "N");

/* calc fuel/pollution data? */
if (FW_Ptr->CALC_FUEL_AND_POLLUTION)
    fprintf(kronos_fp, "%s\n", "Y");
else
    fprintf(kronos_fp, "%s\n", "N");

/* calc measures of effectiveness? */
if (FW_Ptr->CALC_MOES)
    fprintf(kronos_fp, "%s\n", "Y");
else
    fprintf(kronos_fp, "%s\n", "N");

/* output file name */
fprintf(kronos_fp, "%s\n", header->file_name);

/* simulation date */
fprintf(kronos_fp, "%s\n", header->file_date);

/* save instantaneous density? */
if (FW_Ptr->SAVE_INST_DENSITY)
    fprintf(kronos_fp, "%s\n", "Y");
else
    fprintf(kronos_fp, "%s\n", "N");

/* save instantaneous speed? */
if (FW_Ptr->SAVE_INST_SPEED)
    fprintf(kronos_fp, "%s\n", "Y");
else
    fprintf(kronos_fp, "%s\n", "N");

```

```

/* save instantaneous flow rate? */
if (FW_Ptr->SAVE_INST_FLOW)
    fprintf (kronos_fp, "%s\n", "Y");
else
    fprintf (kronos_fp, "%s\n", "N");

/* save average density? */
if (FW_Ptr->SAVE_AVG_DENSITY)
    fprintf (kronos_fp, "%s\n", "Y");
else
    fprintf (kronos_fp, "%s\n", "N");

/* save average speed? */
if (FW_Ptr->SAVE_AVG_SPEED)
    fprintf (kronos_fp, "%s\n", "Y");
else
    fprintf (kronos_fp, "%s\n", "N");

/* save average flow rate? */
if (FW_Ptr->SAVE_AVG_FLOW)
    fprintf (kronos_fp, "%s\n", "Y");
else
    fprintf (kronos_fp, "%s\n", "N");

/* plotting period */
fprintf (kronos_fp, "%d\n", FW_Ptr->Plotting_Period);

/* demand information, one set for each time slice */
for (Time_Index = 0; Time_Index < Num_Time_Slices; Time_Index++)
{
    for (Seg_Index = 0; Seg_Index < Seg_Num; Seg_Index++)
    {
        KSegment = FW_Ptr->Seg[Seg_Index];

        switch (KSegment.Type)
        {
            case PIPELINE:
                break;

            case RHS_SIMPLE_ONRAMP:
            case LHS_SIMPLE_ONRAMP:
            case RHS_1_LANE_ADD_ONRAMP:
            case LHS_1_LANE_ADD_ONRAMP:
            case RHS_SHARED_2_LANE_ONRAMP:
            case LHS_SHARED_2_LANE_ONRAMP:
                KOnRamp = *(KSegment.On_Ramp);
                fprintf (kronos_fp, "%d\n", (int) KOnRamp.Entrance_Rate_At[Time_Index]);
                fprintf (kronos_fp, "%d\n", (int) KOnRamp.Meter_Rate_At[Time_Index]);
                break;

            case RHS_2_LANE_ADD_ONRAMP:
            case LHS_2_LANE_ADD_ONRAMP:
                KOnRamp = *(KSegment.On_Ramp);
                fprintf (kronos_fp, "%d\n", (int) KOnRamp.Entrance_Rate_At[Time_Index]);
                break;

            case RHS_SIMPLE_OFFRAMP:
            case LHS_SIMPLE_OFFRAMP:
            case RHS_1_LANE_DROP_OFFRAMP:
            case LHS_1_LANE_DROP_OFFRAMP:
            case RHS_2_LANE_DROP_OFFRAMP:
            case LHS_2_LANE_DROP_OFFRAMP:
            case RHS_SHARED_2_LANE_OFFRAMP:
            case LHS_SHARED_2_LANE_OFFRAMP:

```

```

KOffRamp = *(KSegment.Off_Ramp);
fprintf(kronos_fp, "%d\n", (int) KOffRamp.Exit_Rate_At[Time_Index]);
/* congested? */
if (KOffRamp.Congested_At[Time_Index])
    fprintf(kronos_fp, "%s\n", "Y");
else
    fprintf(kronos_fp, "%s\n", "N");
fprintf(kronos_fp, "%d\n", (int) KOffRamp.Restricted_Capacity_At[Time_Index]);
fprintf(kronos_fp, "%d\n", (int) KOffRamp.Through_Demand_At[Time_Index]);
fprintf(kronos_fp, "%d\n", (int) KOffRamp.Diversion_Rate_At[Time_Index]);
break;

case RHS_WEAVE:
case LHS_WEAVE:
case USER_NEW_WEAVING:
    KWeaveSeg = *(KSegment.Weaving_Seg);
    KOnRamp = *(KSegment.On_Ramp);
    KOffRamp = *(KSegment.Off_Ramp);
    fprintf(kronos_fp, "%d\n", (int) KOnRamp.Entrance_Rate_At[Time_Index]);
    fprintf(kronos_fp, "%d\n", (int) KOffRamp.Exit_Rate_At[Time_Index]);
    fprintf(kronos_fp, "%d\n", (int) KWeaveSeg.Entrance_Percentage_At[Time_Index]);
    fprintf(kronos_fp, "%d\n", (int) KOnRamp.Meter_Rate_At[Time_Index]);
    fprintf(kronos_fp, "%d\n", (int) KOffRamp.Restricted_Capacity_At[Time_Index]);
    fprintf(kronos_fp, "%d\n", (int) KOffRamp.Through_Demand_At[Time_Index]);
    fprintf(kronos_fp, "%d\n", (int) KOffRamp.Diversion_Rate_At[Time_Index]);
    break;

default:
    break;

    }                /* switch */

    }                /* end for ramp information */

    }                /* end for time index */

return (0);
}
/*..... end TDKPrintFile ..... */

```

```

ushort
TDKEndTranslation (TDKHeader * header)
{
/*.....
* Normal ending of a session. It means that there are no more KRONOS seg-
* ments to be created, so the KRONOS files can be created.
*.....
*/

char file_name[14];
FILE *kronos_fp;
int i;

/* Open KRONOS file and write global variables */
sprintf(file_name, "%s.kr8", header->file_name);
printf("Opening KRONOS file %s.\n", file_name);
if ((kronos_fp = fopen(file_name, "w")) == NULL)
{
    printf("Error opening KRONOS file\n");
    return (-1);
}

TDKPrintFile(header, kronos_fp);

```

```

/* Close KRONOS file */
printf ("Closing KRONOS file %s.\n", file_name);
fclose (kronos_fp);

/*
printf("Printing KRONOS segments...\n");
for(i=0;i<Seg_Num;i++) {
printf("Segment %d is type %d\n",i+1,FW_Ptr->Seg[i].Type);
if(strcmp(FW_Ptr->Seg[i].Name,""))
printf("  Name = %s\n",FW_Ptr->Seg[i].Name);
printf("  Num_Lanes = %d\n",FW_Ptr->Seg[i].Num_Lanes);
printf("  Capacity = %f\n",FW_Ptr->Seg[i].Capacity);
printf("  Length = %d\n",FW_Ptr->Seg[i].Length);
printf("  Init_Flow = %f\n",FW_Ptr->Seg[i].Init_Flow);
printf("  INCIDENT = %d\n",FW_Ptr->Seg[i].INCIDENT);
printf("  INIT_CONG = %d\n",FW_Ptr->Seg[i].INITIALLY_CONGESTED);
}
for(i=0;i<On_Num;i++) {
printf("On Ramp %d for %s\n",i+1,FW_Ptr->On_R[i].Seg->Name);
printf("  Capacity = %f\n",FW_Ptr->On_R[i].Capacity);
printf("  Length = %d\n",FW_Ptr->On_R[i].Length);
printf("  Demand = %f\n",FW_Ptr->On_R[i].Demand);
printf("  Meter_Loc = %d\n",FW_Ptr->On_R[i].Meter_Location);
printf("  METERED = %d\n",FW_Ptr->On_R[i].METERED);
}
for(i=0;i<Off_Num;i++) {
printf("Off Ramp %d for %s\n",i+1,FW_Ptr->On_R[i].Seg->Name);
printf("  Capacity = %f\n",FW_Ptr->Off_R[i].Capacity);
printf("  Length = %d\n",FW_Ptr->Off_R[i].Length);
printf("  Demand = %f\n",FW_Ptr->Off_R[i].Demand);
printf("  CONNECTED = %d\n",FW_Ptr->Off_R[i].CONNECTED);
printf("  RES CAP = %d\n",FW_Ptr->Off_R[i].RESTRICTED_CAPACITY);
}
for(i=0;i<Weave_Num;i++) {
printf("Weave %d for %s\n",i+1,FW_Ptr->On_R[i].Seg->Name);
printf("  Aux Cap 1 = %f\n",FW_Ptr->Weave[i].Aux_Lane_Capacity_1);
printf("  Aux Cap 2 = %f\n",FW_Ptr->Weave[i].Aux_Lane_Capacity_2);
printf("  Aux Demand = %f\n",FW_Ptr->Weave[i].Aux_Lane_Init_Flow);
printf("  Weave length = %d\n",FW_Ptr->Weave[i].Length_Of_Weave);
printf("  Length before = %d\n",FW_Ptr->Weave[i].Length_Before_Weave);
}
for(i=0;i<Lad_Num;i++) {
printf("LAD %d for %s\n",i+1,FW_Ptr->On_R[i].Seg->Name);
printf("  Capacity = %f\n",FW_Ptr->Off_R[i].Capacity);
printf("  Tran Start = %d\n",FW_Ptr->Off_R[i].Length);
printf("  Tran Length = %d\n",FW_Ptr->Off_R[i].Demand);
}
}
/* Free allocated memory */
cfree ((char *) FW_Ptr->On_R);
cfree ((char *) FW_Ptr->Off_R);
cfree ((char *) FW_Ptr->Weave);
cfree ((char *) FW_Ptr->LAD);
cfree ((char *) FW_Ptr->Seg);

return (0);
}
/*..... end TDKEndTranslation ..... */

void
TDKCommunicateError (ushort error)
{
/*****
* Communicates that an error has occurred during the translation session, *
* this error has not been generated by any of the functions contained in this*
*****/

```

```

* file. but by a higher-level function.
* The duty of this function is to abort the translation session, cleaning
* any dynamically allocated data structures.
*****
*/

return;
}
/*..... end TDKCommunicateError .....*/

```

## MAIN.C

```

/* main.c - The KRONOS translator program */

#include <stdio.h>
#include <string.h>

#include "TDFun.h"
#include "kronos.h"

/* non-TDFunction */
void TDAGetNetworkName (int, char *[], char *, ushort *, char[][21]);

int
main (int argc, char *argv[])
{
/*****
* Main function. Tests the TDFunctions.
* Call format: $ tedi network_name view1 ... viewn
*****/
*/

int Kargc;
char *malloc (), *Kargv[2], *temp_ptr;

ushort nbf, nbfiles, nbv, nbn, nbnets, nbviews, nbviewpath_names, error,
nbvehmods, nbe, nbt, i, nbturnings, nbdetectors, nbpoints, nbp, nbentrances,
nbexits, *entrances, *exits, idsections[40], nbnodes, *pidnodes, idnode_added,
nbl, idnode_involved, nbsegments, nbstages, nbs, nbr, nbm, nb, nbd,
nbdet, nbde, idlane, nblanes_acum, nblanes_turning, nbturnings2, nbmessasigns,
nbcontrols, nbc, nblaterals, nbcentroids, nbcenconnections, nbobjects,
nbfiles2, nbf2, flags, nbidsections, idsection_origin, idsection_dest,
nbidsections_intermediate, idsections_intermediate[100], nbidsections_path,
*idsections_path;
int nbcol;
Bool first_time, node_created;
float prob, length, width, factor, acum_flow, f1, f3, f6, f8, f;
double inc;
char last_modified_date[10], network_name[21], view_names[10][21], **view_names_ret,
viewpath_names[10][21], idgen[21], car;
TDHandle handle;
TDName *pTDnames;
TDFile *TDfiles, *TDfile;
TDPoint *polygon, p1, p2, pint, center, *points, po[10], rectangle[4];
TDLineCoef line;
TDSegment *segments, TDsegment;
TDText *pTDtext;
TDNetwork TDnetwork, *pTDnetwork;
TDNetDim *pTDnetdim;
TDView *pTDview;
TDControl *controls;
TDVehMod *pTDvehmods, *pTDvehmod, TDvehmod;
char name[21];
ushort nbresults;
TDResult *results;

```



```

TDResultHeader *pTDheader, TDheader;
TDResultReport report;
ushort nbnetstates;
TDNetState *pTDnetstates;
TDSection section, TDsection;
TDLateral TDlateral, *pTDlateral, *pTDlaterals;
TDDetector *pTDdetectors, *pTDdetector, TDdetector;
TDMetering *pTDMetering, TDMetering;
TDMessaSign *pTDmessasigns, *pTDmessasign, TDmessasign;
TDTurning *pTDturning, *pTDturnings, TDturning;
TDStage *pTDstage, *pTDstages;
TDIdTurning *right, *prio, *pTDidturnings;
TDNode node, TDnode;
TDController TDcontroller, controller;
TDStateReport reportst;
TDStateSection *sectionst;
TDStateEntrance *entrancest;
TDStateVehMod *vehmodst;
TDStateTurning *turningst;
ushort nbvehclasses;
TDVehClass *pTDvehclasses, TDvehclass;
TDControlReport controlreport;
TDControlMessaSign *pTDcmessasigns, *pTDcmessasign, TDcmessasign;
TDBackground background, *backgrounds;
TDCentroid TDcentroid, *pTDcentroid, *pTDcentroids;
TDCenConnection TDCenconnection, *pTDCenconnection, *pTDCenconnections;
TDObjectToCentroid *pTDobjects;
TDIdObject TDidobjects[10];
TDODMatrix *pTDodmatri;
TDODTime TDodtime;
TDKHeader kheader;
FILE *ftemp, *ftemp2;

/* initializes the names of the network and views to consider */
TDAGetNetworkName (argc, argv, network_name, &nbviews, view_names);

/* reads all the file names in the "nets" directory, to determine
the last modified date of the network being considered */
if (error = TDReadDirectoryContents (NULL, &nbfiles, &TDfiles))
goto LERROR;

/* search for file specified by the user to get the date */
strcpy (last_modified_date, "");
for (nbf = 0, TDfile = TDfiles; nbf < nbfiles; nbf++, TDfile++)
{
/* printf("File Name: %s, type: %d, date: %s\n",
TDfile->name, TDfile->type, TDfile->last_modified_date); */
if (!strcmp (network_name, TDfile->name))
{
/* then you've found it */
strcpy (last_modified_date, TDfile->last_modified_date);
break;
}
}
/* end for files */
/* error message should go here if the network name was not found */
*/

/* prints the name of the network, date, and view to open */
printf ("Network = %s\n", network_name);
printf ("Last modified date = %s\n", last_modified_date);

for (nbv = 0; nbv < nbviews; nbv++)
{
if (nbv == 0)

```

```

        {
            printf(" View = ");
        }
    printf("%s", view_names[nbv]);
    if (nbv < (ushort) (nbviews - 1))
        {
            printf(", ");
        }
    }

/* open the network, in ASCII file format */
if (error = TDOpenNetwork (network_name, 2))
    goto LERROR;
/*
    TDAListSections();
    TDAListNodes();
*/

if (nbviews > 0)
    {
        /* restrict the view, meaning restrict the network
            to the sub-network specified */
        /* apparently, it picks the last view for restriction */
        if ((error = TDRestrictView (nbviews - 1, view_names,
            view_names[nbviews - 1])) != 0)
            goto LERROR;
        TDPrintView ();
    }

/* here the testing code ----- */

strcpy (kheader.file_name, network_name);
strcpy (kheader.file_date, "1/30/1996");
/* In future, change this to be dynamic, using the last_modified_date

*/
strcpy (kheader.network_name, network_name);
strcpy (kheader.results_name, "default");
/* In future, this should also be dynamic, user-specified

*/

/* sections along the route */
/* forI494 */
idsection_origin = 42;
idsection_dest = 116;
nbidsections_intermediate = 9;
idsections_intermediate[0] = 75;
idsections_intermediate[1] = 76;
idsections_intermediate[2] = 79;
idsections_intermediate[3] = 83;
idsections_intermediate[4] = 90;
idsections_intermediate[5] = 94;
idsections_intermediate[6] = 101;
idsections_intermediate[7] = 105;
idsections_intermediate[8] = 111;

/* using the origin, destination, and intermediate sections,
    searches for a path connecting the origin and destination,
    using a "closest angle" approach */
if (error = TDFindRoutePath (idsection_origin, idsection_dest,
    nbidsections_intermediate, idsections_intermediate,
    &nbidsections_path, &idsections_path))

    goto LERROR;
printf ("Path through network successfully found.\n");

```

```

/* the number of the sections along the path */
kheader.nbidsections = nbidsections_path;
/* allocate memory for the array of section ids */
if (kheader.nbidsections > 0)
{
    if ((kheader.idsections = (ushort *)
        TDcalloc ((size_t) kheader.nbidsections,
            sizeof (ushort))) == NULL)
    {
        error = MemAlloc;
        goto LERROR;
    }
}

for (nbs = 0; nbs < kheader.nbidsections; nbs++)
{
    kheader.idsections[nbs] = idsections_path[nbs];
}

/* open the desired results container */
if (error = TDOpenResult (0, NULL,
    kheader.results_name, &handle, &report))
    goto LERROR;

/* read the vehicle modalities, and use it to assign percentages */
if (error = TDReadVehMods (handle, &nbvehmods, &pTDvehmods))
    goto LERROR;
kheader.nbvehmods = nbvehmods;

/* read the network state (time slice data) names and time offsets
that exist for the results container specified */
if (error = TDReadNetStates (&nbfiles, &pTDnetstates))
    goto LERROR;
kheader.nbnetstates = nbfiles;
printf ("Number of time slices = %d.\n", kheader.nbnetstates);
if (nbfiles > 0)
{
    /* allocate memory for the network state names */
    if ((kheader.netstates = (TDName *) TDcalloc ((size_t) nbfiles,
        sizeof (TDName))) == NULL)
    {
        error = MemAlloc;
        goto LERROR;
    }
}

/* fill array of network state names */
for (nbf = 0; nbf < nbfiles; nbf++)
{
    strcpy (kheader.netstates[nbf], pTDnetstates[nbf].name);
}

kheader.time_offset = (int) ((pTDnetstates[1].time_offset
    - pTDnetstates[0].time_offset) / 60.0);
printf ("Time slice duration = %d minutes.\n", kheader.time_offset);

if (error = TDCloseResult (0))
    goto LERROR;

if (error = TDTranslateToKRONOS (&kheader))
    goto LERROR;

/* close the network when finished */
if (error = TDCloseNetwork (0))
    goto LERROR;

```

```

#ifdef UNIX
    exit (0);
#else
    return (0);
#endif

/* prints the error message */
LERROR:
    TDTxtMessage (error);
    printf ("ERROR- %s\n", TDMMessage);

#ifdef UNIX
    exit (1);
#else
    return (1);
#endif
}
/*..... end of the main function ..... */

```

## MAKEFILE.C

```

#
# Makefile for the TDFunctions with the added KRONOS translation TDKFunctions
#
#
# Options : all , tedi, library
#
#
# significant suffixes
.SUFFIXES : .o .h .c

EXEHOME=../nets

# compiler macro
#CC = gcc -g -I/usr/lang/SC2.0.1/include/cc_411

CC = cc -g -woff 3262
#CC = acc -Xa -g -fnonstd
#CC = /usr/lang/SC1.0/acc -Xa -g -fnonstd

#CC = gcc -g
# library archive macro
AR = ar rcv libtd.a

# libraries needed
LIBS = -lm

# H files included in all C files
HFILES = itypes.h ifuncti.h iglobal.h imacros.h TDIFun.h \
        TDSwitch.h TDTypes.h TDFuncti.h TDErrors.h TDGlobal.h TDFun.h \
        kronos.h

# C files in the project
CFILES = main.c inames.c imem.c istrio.c imessage.c \
        iview.c view.c vehmodli.c polyseg.c \
        iobject.c iblock.c itext.c ivehc.c ignessa.c isection.c \
        inode.c icontll.c icentroi.c icross.c \
        network.c net.c block.c text.c vehc.c gnessa.c section.c \
        lateral.c selper.c detector.c meter.c vms.c \
        node.c junctu.c juncti.c turning.c stage.c \
        backgrou.c contll.c condev.c centroi.c cencon.c \
        control.c cjuncti.c route.c cmeter.c cvms.c icontrol.c \

```

```

result.c  ireult.c  vehmod.c  state.c  istate.c \
ssection.c  odmatri.c  iodmatri.c  odsta.c \
geo.c  gretwork.c  gblock.c  gtext.c  gvehc.c \
ggmessa.c  gsection.c  gnode.c  gcontll.c  gcentroi.c \
anetwork.c  ablock.c  atext.c  avehc.c \
agmessa.c  asection.c  anode.c  acontll.c  acentroi.c \
translat.c  kronos.c  ikronos.c  kinit.c

```

```
OBJFILES = $(CFILES:.c=.o)
```

```
# if a .h file has changed then compile all
$(EXEHOME)tdtest:: $(HFILES)
    $(CC) -c $(CFILES)
```

```
# tedi depends on all the object files
$(EXEHOME)tdtest:: $(OBJFILES)
    $(CC) -o $@ $(OBJFILES) $(LIBS)
```

```
# rule to obtain the .o from the .c
.c.o:
    $(CC) -c $<
```

```
all:
    touch $(HFILES) $(CFILES)
    @make
```

```
library: $(OBJFILES)
    $(AR) $(OBJFILES)
# a 'ranlib libtd.a' command is necessary thereafter
```

## TRANSLAT.C

```
/* translat.c - functions for translating GETRAM sections into KRONOS segments */
```

```
#include <limits.h>
#include <string.h>          /* memset() */
```

```
#include "TDFun.h"
#include "TDIFun.h"
```

```
/* this is the main file for results */
#define KRONOS_MAIN
#include "kronos.h"
```

```
ushort
TDTranslateToKRONOS (TDKHeader * header)
{
/******
 * Given a header with the file name, control plan to use and network *
 * states, translates it to KRONOS format. *
 * Errors: *
 * KRONOSFirstSectionUnk - couldn't find the first section required by*
 * KRONOS. *
*****
*/
ushort error, type, idsection, nbturnings, nbt, nbs, nbsections, idsection_prev,
idsection_next;
Bool end_vsegments;
TDHandle handle;
TDResultReport report;
```

```

TDKSegment segment;
TDSection TDsection, TDsection_next;
TDTurning *TDturnings, *TDturning;
TDStateSection *ssections;

if (header == NULL)
    return (ParamError);

/* open the desired results container */
if (error = TDOpenResult (0, NULL, header->results_name, &handle, &report))
    return (error);

/* get time-sliced sections state */
if (error = TDRReadStateSectionsTimeSliced (header->nbnetstates,
                                             header->netstates, &nsections, &ssections))
    return (error);

/* we can already close the results container */
if (error = TDCloseResult (0))
    return (error);

/* to ease debugging, clear the vsegments array */
TDKVClearVSegments ((ushort) TDKMAXVSEGMENTS, vsegments);

/* in the vsegments[], allocate enough space for the netstates
 * entrance flows in each vsegment
 */
if (error = TDKVAllocateFlows ((ushort) TDKMAXVSEGMENTS, vsegments,
                               header->nbnetstates))
    return (error);

/* to ease debugging */
nbnetstates = header->nbnetstates;

/* prepare TDKSegment structure */
if (error = TDKIPrepareSegment (header, &segment))
    return (error);

/* just at the beginning ----- */
if (error = TDKBeginTranslation (header))
    goto LERROR_KRONOS;

/* loop along the route, creating the intermediate vsegments[] array */
nbvsegments = 0;
idsection_prev = 0;
for (nbs = 0; nbs < header->nbidsections; nbs++)
{
    idsection = header->idsections[nbs];
    end_vsegments = (nbs == (header->nbidsections - 1));
    if (end_vsegments)
    {
        idsection_next = 0;
    }
    else
    {
        idsection_next = header->idsections[nbs + 1];
    }

    /* decompose this section in vsegments, adding them to the end
     * of the vsegments[] array
     */
    if (error = TDKVDecomposeSection (idsection, header,
                                     idsection_prev, idsection_next,
                                     nsections, ssections,

```

```

        (ushort) TDKMAXVSEGMENTS, &nbvsegments, vsegments))
    goto LERROR;

/* tries to find a KRONOS segment, from the beginning. When found, code -- */

/* calls to the KRONOS-specific functions and deletes these vsegments.
 * shifting the contents of the vsegments[] array
 */
if (error = TDKVTreatmentForKRONOS (header, &segment,
                                   (ushort) TDKMAXVSEGMENTS,
                                   &nbvsegments, vsegments, end_vsegments))
    goto LERROR;

    idsection_prev = idsection;
}

if (nbvsegments > 0)
{
    /* reached the end but some sections could not be translated */
    /* print vsegments to screen */
    TDKVPrintVSegments ();

    error = KRONOSSectionTransNot;
    goto LERROR;
}

/* at the end */
if (error = TDKEndTranslation (header))
    goto LERROR_KRONOS;

TDKIDeallocateSegment (&segment);

return (0);

/* errors ----- */
LERROR_KRONOS:
    return (error);
LERROR:
    /* notify to the KRONOS translator that there has been an error, so
     * that it should interrupt the current translation session
     */
    TDKCommunicateError (error);
    return (error);
}
/*:..... end TDTranslateToKRONOS ..... */

ushort
TDKVTreatmentForKRONOS (TDKHeader * header, TDKSegment * segment,
                       ushort maxvsegments, ushort * nbvsegments,
                       TDKVSegment * vsegments, Bool end_vsegments)
{
    /*
     * Examines the array vsegments[] from the start, trying to find a KRONOS
     * segment. When found, calls to the KRONOS-specific functions and deletes
     * these vsegments, shifting the contents of the vsegments[] array.
     * The structure segment has just been correctly allocated, nothing more.
     */
    ushort error, nbvsegments_affected, idv, idvsegment_cannotdecide, nbvseg;
    Bool cannot_decide_yet, evaluate_only, end_vsegments_int;

    if (*nbvsegments == 0)

```

```

return (0);

while (TRUE)
{
    nbvsegments_affected = 0;
    idvsegment_cannotdecide = *nbvsegments;      /* just beyond the end */
    cannot_decide_yet = FALSE;
    end_vsegments_int = end_vsegments;
    /* look for each possible KRONOS segment: begin by the most compli-
    * cated cases: the ones that involve more vsegments. This is a
    * must, otherwise they would always be considered as simpler
    * segments.
    */
    /*
    if (nbvsegments_affected == 0 && !cannot_decide_yet) {
    if (error = TDKVTreatmentK16(header, segment,
    *nbvsegments, vsegments, end_vsegments,
    &nbvsegments_affected, &cannot_decide_yet))
    return(error);
    }
    */
    if (nbvsegments_affected == 0 && !cannot_decide_yet)
    {
        for (idv = 0; idv < idvsegment_cannotdecide; idv++)
        {
            /* examines 3 or less segments */
            if ((int) (idvsegment_cannotdecide - idv) < 3)
            {
                nbvseg = (idvsegment_cannotdecide - idv);
            }
            else
            {
                nbvseg = 3;
            }
            evaluate_only = TRUE;
            if (error = TDKVTreatment3V (header, segment, evaluate_only,
            nbvseg, vsegments + idv, end_vsegments_int,
            &nbvsegments_affected, &cannot_decide_yet))
                return (error);
            if (nbvsegments_affected > 0 || cannot_decide_yet)
            {
                /* has found a 3V segment here! */
                if (idv == 0 && !cannot_decide_yet)
                {
                    evaluate_only = FALSE;
                    if (error = TDKVTreatment3V (header, segment, evaluate_only,
                    nbvseg, vsegments + idv, end_vsegments_int,
                    &nbvsegments_affected, &cannot_decide_yet))
                        return (error);
                }
            }
            else
            {
                /* it is not at the beginning of vsegments */
                if (nbvsegments_affected > 0)
                {
                    /* bit it has really found a KRONOS segment at that
                    * point. The others should not think that there
                    * might be something else further
                    */
                    end_vsegments_int = TRUE;
                }
                idvsegment_cannotdecide = idv;
                nbvsegments_affected = 0;
                cannot_decide_yet = FALSE;
            }
        }
    }
}

```



```

        break;
    }
}

if (nbvsegments_affected == 0 && !cannot_decide_yet)
{
    for (idv = 0; idv < idvsegment_cannotdecide; idv++)
    {
        /* examines 2 or less segments */
        if ((int) (idvsegment_cannotdecide - idv) < 2)
        {
            nbvseg = (idvsegment_cannotdecide - idv);
        }
        else
        {
            nbvseg = 2;
        }
        evaluate_only = TRUE;
        if (error = TDKVTreatment2V_1 (header, segment, evaluate_only,
            nbvseg, vsegments + idv, end_vsegments_int,
            &nbvsegments_affected, &cannot_decide_yet))
            return (error);
        if (nbvsegments_affected > 0 || cannot_decide_yet)
        {
            /* has found a 2V segment here! */
            if (idv == 0 && !cannot_decide_yet)
            {
                evaluate_only = FALSE;
                if (error = TDKVTreatment2V_1 (header, segment, evaluate_only,
                    nbvseg, vsegments + idv, end_vsegments_int,
                    &nbvsegments_affected, &cannot_decide_yet))
                    return (error);
            }
            else
            {
                /* it is not at the beginning of vsegments */
                if (nbvsegments_affected > 0)
                {
                    /* bit it has really found a KRONOS segment at that
                     * point. The others should not think that there
                     * might be something else further
                     */
                    end_vsegments_int = TRUE;
                }
                idvsegment_cannotdecide = idv;
                nbvsegments_affected = 0;
                cannot_decide_yet = FALSE;
            }
            break;
        }
    }
}

if (nbvsegments_affected == 0 && !cannot_decide_yet)
{
    for (idv = 0; idv < idvsegment_cannotdecide; idv++)
    {
        /* examines 2 or less segments */
        if ((int) (idvsegment_cannotdecide - idv) < 2)
        {
            nbvseg = (idvsegment_cannotdecide - idv);
        }
        else
        {
            nbvseg = 2;

```

```

    }
    evaluate_only = TRUE;
    if (error = TDKVTreatment2V_2 (header, segment, evaluate_only,
        nbvseg, vsegments + idv, end_vsegments_int,
        &nbvsegments_affected, &cannot_decide_yet))
        return (error);
    if (nbvsegments_affected > 0 || cannot_decide_yet)
    {
        /* has found a 2V segment here! */
        if (idv == 0 && !cannot_decide_yet)
        {
            evaluate_only = FALSE;
            if (error = TDKVTreatment2V_2 (header, segment, evaluate_only,
                nbvseg, vsegments + idv, end_vsegments_int,
                &nbvsegments_affected, &cannot_decide_yet))
                return (error);
        }
        else
        {
            /* it is not at the beginning of vsegments */
            if (nbvsegments_affected > 0)
            {
                /* bit it has really found a KRONOS segment at that
                 * point. The others should not think that there
                 * might be something else further
                 */
                end_vsegments_int = TRUE;
            }
            idvsegment_cannotdecide = idv;
            nbvsegments_affected = 0;
            cannot_decide_yet = FALSE;
        }
        break;
    }
}

if (nbvsegments_affected == 0 && !cannot_decide_yet)
{
    for (idv = 0; idv < idvsegment_cannotdecide; idv++)
    {
        /* examines 1 or less segments */
        if ((int) (idvsegment_cannotdecide - idv) < 1)
        {
            nbvseg = (idvsegment_cannotdecide - idv);
        }
        else
        {
            nbvseg = 1;
        }
        evaluate_only = TRUE;
        if (error = TDKVTreatment1V (header, segment, evaluate_only,
            nbvseg, vsegments + idv, end_vsegments_int,
            &nbvsegments_affected, &cannot_decide_yet))
            return (error);
        if (nbvsegments_affected > 0 || cannot_decide_yet)
        {
            /* has found a 1V segment here! */
            if (idv == 0 && !cannot_decide_yet)
            {
                evaluate_only = FALSE;
                if (error = TDKVTreatment1V (header, segment, evaluate_only,
                    nbvseg, vsegments + idv, end_vsegments_int,
                    &nbvsegments_affected, &cannot_decide_yet))
                    return (error);
            }
        }
    }
}

```

```

else
{
/* it is not at the beginning of vsegments */
if (nbvsegments_affected > 0)
{
/* bit it has really found a KRONOS segment at that
* point. The others should not think that there
* might be something else further
*/
end_vsegments_int = TRUE;
}
idvsegment_cannotdecide = idv;
nbvsegments_affected = 0;
cannot_decide_yet = FALSE;
}
break;
}
}

if ((!cannot_decide_yet) && nbvsegments_affected == 0 &&
idvsegment_cannotdecide > 0)
{
/* reached the end but some sections could not be translated */
error = KRONOSSectionTransNot;
return (error);
}

/* shifts the vsegments array to the left */
if (error = TDKVShiftLeftVSegments (maxvsegments, nbvsegments,
vsegments, nbvsegments_affected))
return (error);
if (*nbvsegments == 0 || cannot_decide_yet ||
idvsegment_cannotdecide == 0)
break;
}

return (0);
}
/*:..... end TDKVTreatmentForKRONOS .....*/

```

```

ushort
TDKVDecomposeSection (ushort idsection, TDKHeader * header,
ushort idsection_prev, ushort idsection_next,
ushort nbsections, TDStateSection * ssections,
ushort maxvsegments, ushort * nbvsegments, TDKVSegment * vsegments)
{
/*****
* Given a section, decomposes it in 1 to maximum 3 vsegments. It adds
* these segments to the end of the array vsegments[].
* Divides the section in a vsegment corresponding to the entrance lateral, a
* 'main vsegment' between the entrance and exit laterals, and another
* vsegment corresponding to the exit lateral.
*
* Errors:
* KRONOSSectionTransNot: Cannot translate a certain section
* KRONOSTrans2L: Cannot translate this section. There are two late-
* rals at the same section entrance or exit
* KRONOSVSegmentsInsuf: Reached the limit of internal virtual
* segments.
* KRONOSJunction: Cannot translate this section. There is a junction
* at its entrance or exit.
*****/

```

```

*   KRONOS2tPlus: Cannot translate this section. There is more than one *
*   entrance or exit.
*   KRONOS2lanesPlus: Cannot translate this section. An entrance or exit*
*   section has more than 2 lanes.
*****
*/
ushort nbl, nblaterals, lenrance_type, lexit_type, nbsegs, nbs, nbturmings,
nbt, node_entrance_idsection, node_dest_exit_idsection, node_entrance_type,
node_dest_exit_type, node_entrance_nblanes, node_dest_exit_nblanes,
nbnetsstates, nbvehmods, error;
Bool main_exist, lenrance_exist, lexit_exist, node_entrance_exist, node_dest_exit_exist;
float section_length, section_width, main_length, lenrance_length, lexit_length,
dist_begin, dist_end;
TDSection TDsection, TDsection_entrance, TDsection_exit;
TDLateral *TDlaterals, *TDlateral;
TDTurning *TDturnings, *TDturning;
TDKVSegment *vsegment;

nbnetsstates = header->nbnetsstates;
nbvehmods = header->nbvehmods;

if (error = TDReadSection (idsection, &TDsection))
return (error);

/* find section length */
if (error = TDGetSectionDim (TDsection.trapezoid, &section_length,
&section_width))
return (error);

if (error = TDReadSectionLaterals (idsection, &nblaterals,
&TDlaterals))
return (error);
/* initialize to empty */
main_exist = lenrance_exist = lexit_exist = FALSE;
lenrance_type = lexit_type = 0;
main_length = lenrance_length = lexit_length = 0;
nbsegs = 0;

/* suppose that there are no laterals, just main segment */
main_exist = TRUE;
main_length = section_length;
nbsegs = 1;

/* set laterals type and length */
for (nbl = 0; nbl < nblaterals; nbl++)
{
TDlateral = &(TDlaterals[nbl]);

if (TDlateral->type == 1 || TDlateral->type == 2)
{
/* entrance lateral */
/* check for 2 laterals at the same entrance or exit */
if (lenrance_exist)
return (KRONOSTrans2L);

lenrance_exist = TRUE;
lenrance_type = TDlateral->type;
lenrance_length = TDlateral->length;
main_length += lenrance_length;
nbsegs += 1;
}
else
{
/* exit lateral */
/* check for 2 laterals at the same entrance or exit */

```

```

        if (lexit_exist)
            return (KRONOSTrans2L);

        lexit_exist = TRUE;
        lexit_type = TDlateral->type;
        lexit_length = TDlateral->length;
        main_length -= lexit_length;
        nbsegs += 1;
    }
}

/* look if the main segment has to be suppressed due to its short
 * length
 */
if (main_length < 30 && (lentrance_exist || lexit_exist))
{
    /* here special care has to be taken, because it may be possible
     * that the entrance and exit laterals, although at different
     * sides of the section, superimpose over the section.
     * In that case, main_length can be very < 0
     */
    /* suppress it */
    if (lentrance_exist)
    {
        /* if possible, add it to the entrance lateral */
        letrance_length += main_length;
        /* special care */
        if (lentrance_length < 15)
        {
            letrance_length = 15;
        }
    }
    else
    {
        lexit_length += main_length;
        /* special care */
        if (lexit_length < 15)
        {
            lexit_length = 15;
        }
    }
    main_exist = FALSE;
    main_length = 0;
    nbsegs -= 1;
}

/* check that there is enough space in the array */
if (((int) (*nbvsegments + nbsegs) > (int) maxvsegments)
{
    return (KRONOSVSegmentsInsuf);
}

/* look for the characteristics of the origin node */
if (idsection_prev != 0)
{
    node_origin_exist = (TDsection.idnode_origin != 0);
}
else
{
    node_origin_exist = FALSE;
}
node_origin_entrance_exist = FALSE;
node_origin_entrance_idsection = 0;
node_origin_entrance_type = 0;
node_origin_entrance_nblanes = 0;

```

```

if (node_origin_exist)
{
    /* make sure that it is a juncture! */
    if (!TDsection.juncture_origin)
        return (KRONOSJunction);

    /* look if there is an entrance */
    if (error = TDReadPrevTurnings (idsection, &nbtturnings,
                                    &TDturnings))
        return (error);
    /* KRONOS cannot accept more than an entrance section */
    if (nbtturnings > 2)
        return (KRONOS2tPlus);

    /* search for the one that is diferent from idsection_prev */
    for (nbt = 0; nbt < nbtturnings; nbt++)
    {
        TDturning = TDturnings + nbt;

        if (TDturning->idsection_origin != idsection_prev)
        {
            /* got it! */
            if (error = TDReadSection (TDturning->idsection_origin,
                                      &TDsection_entrance))
                return (error);
            node_origin_entrance_exist = TRUE;
            node_origin_entrance_idsection = TDturning->idsection_origin;
            if (nbt == 0)
            {
                node_origin_entrance_type = 1;
            }
            else
            {
                node_origin_entrance_type = 2;
            }
            node_origin_entrance_nblanes = TDsection_entrance.nblanes;
            if (node_origin_entrance_nblanes > 2)
            {
                return (KRONOS2lanesPlus);
            }
        }
    }
}

/* look for the characteristics of the dest node */
if (idsection_next != 0)
{
    node_dest_exist = (TDsection.idnode_dest != 0);
}
else
{
    node_dest_exist = FALSE;
}
node_dest_exit_exist = FALSE;
node_dest_exit_idsection = 0;
node_dest_exit_type = 0;
node_dest_exit_nblanes = 0;
if (node_dest_exist)
{
    /* make sure that it is a juncture! */
    if (!TDsection.juncture_dest)
        return (KRONOSJunction);

    /* look if there is an entrance */
    if (error = TDReadTurnings (idsection, &nbtturnings,
                                &TDturnings))

```

```

        return (error);
/* KRONOS cannot accept more than an exit section */
if (nbturnings > 2)
    return (KRONOS2tPlus);

/* search for the one that is different from idsection_next */
for (nbt = 0; nbt < nbturnings; nbt++)
    {
        TDturning = TDturnings + nbt;

        if (TDturning->idsection_dest != idsection_next)
            {
                /* got it! */
                if (error = TDRReadSection (TDturning->idsection_dest,
                    &TDsection_exit))
                    return (error);
                node_dest_exit_exist = TRUE;
                node_dest_exit_idsection = TDturning->idsection_dest;
                if (nbt == 0)
                    {
                        node_dest_exit_type = 1;
                    }
                else
                    {
                        node_dest_exit_type = 2;
                    }
                node_dest_exit_nblanes = TDsection_exit.nblanes;
                if (node_dest_exit_nblanes > 2)
                    {
                        return (KRONOS2lanesPlus);
                    }
            }
    }
}

/* put these vsegments in the array */
nbs = 0;
dist_begin = 0;
/* lateral-entrance vsegment */
if (lentrance_exist)
    {
        vsegment = vsegments + *nbvsegments + nbs;
        nbs += 1;
        vsegment->idsection = idsection;
        vsegment->length = lentrance_length;

        vsegment->dist_begin = dist_begin;
        vsegment->dist_end = vsegment->dist_begin + vsegment->length;
        dist_begin = vsegment->dist_end;

        vsegment->nblanes = TDsection.nblanes;
        TDKVTransferFlows (vsegment->idsection,
            nbnetstates, nbvehmods, nbsections, ssections,
            vsegment->flows);

        vsegment->lateral_type = lentrance_type;

        vsegment->node_origin_exist = node_origin_exist;
        vsegment->node_origin_entrance_exist = node_origin_entrance_exist;
        vsegment->node_origin_entrance_idsection =
            node_origin_entrance_idsection;
        vsegment->node_origin_entrance_type =
            node_origin_entrance_type;
        vsegment->node_origin_entrance_nblanes =
            node_origin_entrance_nblanes;
    }
}

```

```

if(vsegment->node_origin_entrance_idsection != 0)
{
    TDKVTransferFlows (vsegment->node_origin_entrance_idsection,
                      nbnetstates, nbvehmods, nbsections, ssections,
                      vsegment->node_origin_entrance_flows);
}

if ((!main_exist) && (!lexit_exist))
{
    vsegment->node_dest_exist = node_dest_exist;
    vsegment->node_dest_exit_exist = node_dest_exit_exist;
    vsegment->node_dest_exit_idsection =
        node_dest_exit_idsection;
    vsegment->node_dest_exit_type =
        node_dest_exit_type;
    vsegment->node_dest_exit_nblanes =
        node_dest_exit_nblanes;
    if (vsegment->node_dest_exit_idsection != 0)
    {
        TDKVTransferFlows (vsegment->node_dest_exit_idsection,
                          nbnetstates, nbvehmods, nbsections, ssections,
                          vsegment->node_dest_exit_flows);
    }
}

}

/* main vsegment */
if (main_exist)
{
    vsegment = vsegments + *nbvsegments + nbs;
    nbs += 1;
    vsegment->idsection = idsection;
    vsegment->length = main_length;

    vsegment->dist_begin = dist_begin;
    vsegment->dist_end = vsegment->dist_begin + vsegment->length;
    dist_begin = vsegment->dist_end;

    vsegment->nblanes = TDsection.nblanes;
    TDKVTransferFlows (vsegment->idsection,
                      nbnetstates, nbvehmods, nbsections, ssections,
                      vsegment->flows);

    vsegment->lateral_type = 0;

    if (!entrance_exist)
    {
        vsegment->node_origin_exist = node_origin_exist;
        vsegment->node_origin_entrance_exist = node_origin_entrance_exist;
        vsegment->node_origin_entrance_idsection =
            node_origin_entrance_idsection;
        vsegment->node_origin_entrance_type =
            node_origin_entrance_type;
        vsegment->node_origin_entrance_nblanes =
            node_origin_entrance_nblanes;
        if (vsegment->node_origin_entrance_idsection != 0)
        {
            TDKVTransferFlows (vsegment->node_origin_entrance_idsection,
                              nbnetstates, nbvehmods, nbsections, ssections,
                              vsegment->node_origin_entrance_flows);
        }
    }
}

if (!lexit_exist)
{
    vsegment->node_dest_exist = node_dest_exist;
    vsegment->node_dest_exit_exist = node_dest_exit_exist;

```



```

vsegment->node_dest_exit_idsection =
node_dest_exit_idsection;
vsegment->node_dest_exit_type =
node_dest_exit_type;
vsegment->node_dest_exit_nblanes =
node_dest_exit_nblanes;
if (vsegment->node_dest_exit_idsection != 0)
{
    TDKVTransferFlows (vsegment->node_dest_exit_idsection,
                      nbnetstates, nbvehmods, nbsections, ssections,
                      vsegment->node_dest_exit_flows);
}
}

/* lateral-exit vsegment */
if (lexit_exist)
{
    vsegment = vsegments + *nbvsegments + nbs;
    nbs += 1;
    vsegment->idsection = idsection;
    vsegment->length = lexit_length;

    vsegment->dist_begin = dist_begin;
    vsegment->dist_end = vsegment->dist_begin + vsegment->length;
    dist_begin = vsegment->dist_end;

    vsegment->nblanes = TDsection.nblanes;
    TDKVTransferFlows (vsegment->idsection,
                      nbnetstates, nbvehmods, nbsections, ssections,
                      vsegment->flows);

    vsegment->lateral_type = lexit_type;

    if ((!main_exist) && (!lentrance_exist))
    {
        vsegment->node_origin_exist = node_origin_exist;
        vsegment->node_origin_entrance_exist = node_origin_entrance_exist;
        vsegment->node_origin_entrance_idsection =
node_origin_entrance_idsection;
        vsegment->node_origin_entrance_type =
node_origin_entrance_type;
        vsegment->node_origin_entrance_nblanes =
node_origin_entrance_nblanes;
        if (vsegment->node_origin_entrance_idsection != 0)
        {
            TDKVTransferFlows (vsegment->node_origin_entrance_idsection,
                              nbnetstates, nbvehmods, nbsections, ssections,
                              vsegment->node_origin_entrance_flows);
        }
    }

    vsegment->node_dest_exist = node_dest_exist;
    vsegment->node_dest_exit_exist = node_dest_exit_exist;
    vsegment->node_dest_exit_idsection =
node_dest_exit_idsection;
    vsegment->node_dest_exit_type =
node_dest_exit_type;
    vsegment->node_dest_exit_nblanes =
node_dest_exit_nblanes;
    if (vsegment->node_dest_exit_idsection != 0)
    {
        TDKVTransferFlows (vsegment->node_dest_exit_idsection,
                          nbnetstates, nbvehmods, nbsections, ssections,
                          vsegment->node_dest_exit_flows);
    }
}

```

```

    }

    *nbvsegments += nbsegs;

    return (0);
}
/*..... end TDKVDecomposeSection .....*/

ushort
TDKVTreatment1V (TDKHeader * header, TDKSegment * segment, Bool evaluate_only,
                ushort nbvsegments, TDKVSegment * vsegments,
                Bool end_vsegments, ushort * nbsegments_affected,
                Bool * cannot_decide_yet)
{
    /*.....
    * Looks for a KRONOS type 1 - .. segment (all that require just one
    * virtual segment) from the beginning of the vsegments[] array.
    * Specifically, now it considers these types: 1-11, 17-20.
    * If found, builds the appropriate call to TDKCreateSegment().
    *.....*/
    ushort error, type;
    float *flows, section_length, section_width;
    TDKVSegment *vsegment;
    TDKUnit *unit;
    TDKSection *ksection;

    *nbsegments_affected = 0;
    *cannot_decide_yet = FALSE;

    if (nbvsegments == 0)
        return (0);

    /* consider just the first segment */
    vsegment = &(vsegments[0]);

    if ((vsegment->lateral_type == 0) &&
        (!vsegment->node_origin_entrance_exist) &&
        (!vsegment->node_dest_exit_exist))
    {
        /* no laterals, no entrance and no exit */
        type = 1;
    }
    else if ((vsegment->lateral_type == 1) &&
             (vsegment->node_origin_entrance_exist) &&
             (vsegment->node_origin_entrance_type == 1) &&
             (vsegment->node_origin_entrance_nlanes == 1) &&
             (!vsegment->node_dest_exit_exist))
    {
        /* right-entrance lateral, entrance on the right-most lane and
        * 1 lane, and no exit
        */
        type = 2;
    }
    else if ((vsegment->lateral_type == 3) &&
             (!vsegment->node_origin_entrance_exist) &&
             (vsegment->node_dest_exit_exist) &&
             (vsegment->node_dest_exit_type == 1) &&
             (vsegment->node_dest_exit_nlanes == 1))
    {
        /* right-exit lateral, no entrance, and exit on the right-most lane
        * and 1 lane
        */

```

```

    type = 3;
}
else if ((vsegment->lateral_type == 0) &&
        (vsegment->node_origin_entrance_exist) &&
        (vsegment->node_origin_entrance_type == 1) &&
        (vsegment->node_origin_entrance_nblanes == 1) &&
        (vsegment->node_dest_exit_exist) &&
        (vsegment->node_dest_exit_type == 1) &&
        (vsegment->node_dest_exit_nblanes == 1))
{
    /* no laterals, entrance on the right-most lane and 1 lane,
     * and exit on the right-most lane and 1 lane
     */
    type = 4;
}
else if ((vsegment->lateral_type == 2) &&
        (vsegment->node_origin_entrance_exist) &&
        (vsegment->node_origin_entrance_type == 2) &&
        (vsegment->node_origin_entrance_nblanes == 1) &&
        (!vsegment->node_dest_exit_exist))
{
    /* left-entrance lateral, entrance on the left-most lane and
     * 1 lane, and no exit
     */
    type = 5;
}
else if ((vsegment->lateral_type == 4) &&
        (!vsegment->node_origin_entrance_exist) &&
        (vsegment->node_dest_exit_exist) &&
        (vsegment->node_dest_exit_type == 2) &&
        (vsegment->node_dest_exit_nblanes == 1))
{
    /* left-exit lateral, no entrance, and exit on the left-most lane
     * and 1 lane
     */
    type = 6;
}
else if ((vsegment->lateral_type == 0) &&
        (vsegment->node_origin_entrance_exist) &&
        (vsegment->node_origin_entrance_type == 2) &&
        (vsegment->node_origin_entrance_nblanes == 1) &&
        (vsegment->node_dest_exit_exist) &&
        (vsegment->node_dest_exit_type == 2) &&
        (vsegment->node_dest_exit_nblanes == 1))
{
    /* no laterals, entrance on the left-most lane and 1 lane,
     * and exit on the left-most lane and 1 lane
     */
    type = 7;
}
else if ((vsegment->lateral_type == 0) &&
        (vsegment->node_origin_entrance_exist) &&
        (vsegment->node_origin_entrance_type == 1) &&
        (vsegment->node_origin_entrance_nblanes == 1) &&
        (!vsegment->node_dest_exit_exist))
{
    /* no laterals, entrance on the right-most lane and
     * 1 lane, and no exit
     */
    type = 8;
}
else if ((vsegment->lateral_type == 0) &&
        (!vsegment->node_origin_entrance_exist) &&
        (vsegment->node_dest_exit_exist) &&
        (vsegment->node_dest_exit_type == 1) &&
        (vsegment->node_dest_exit_nblanes == 1))

```

```

{
/* no laterals, no entrance, and exit on the right-most lane
 * and 1 lane
 */
type = 9;
}
else if ((vsegment->lateral_type == 0) &&
        (vsegment->node_origin_entrance_exist) &&
        (vsegment->node_origin_entrance_type == 2) &&
        (vsegment->node_origin_entrance_nblanes == 1) &&
        (!vsegment->node_dest_exit_exist))
{
/* no laterals, entrance on the left-most lane and
 * 1 lane, and no exit
 */
type = 10;
}
else if ((vsegment->lateral_type == 0) &&
        (!vsegment->node_origin_entrance_exist) &&
        (vsegment->node_dest_exit_exist) &&
        (vsegment->node_dest_exit_type == 2) &&
        (vsegment->node_dest_exit_nblanes == 1))
{
/* no laterals, no entrance, and exit on the left-most lane
 * and 1 lane
 */
type = 11;
}
else if ((vsegment->lateral_type == 0) &&
        (vsegment->node_origin_entrance_exist) &&
        (vsegment->node_origin_entrance_type == 1) &&
        (vsegment->node_origin_entrance_nblanes == 2) &&
        (!vsegment->node_dest_exit_exist))
{
/* no laterals, entrance on the right-most lane and
 * 2 lanes, and no exit
 */
type = 17;
}
else if ((vsegment->lateral_type == 0) &&
        (!vsegment->node_origin_entrance_exist) &&
        (vsegment->node_dest_exit_exist) &&
        (vsegment->node_dest_exit_type == 1) &&
        (vsegment->node_dest_exit_nblanes == 2))
{
/* no laterals, no entrance, and exit on the right-most lane
 * and 2 lanes
 */
type = 18;
}
else if ((vsegment->lateral_type == 0) &&
        (vsegment->node_origin_entrance_exist) &&
        (vsegment->node_origin_entrance_type == 2) &&
        (vsegment->node_origin_entrance_nblanes == 2) &&
        (!vsegment->node_dest_exit_exist))
{
/* no laterals, entrance on the left-most lane and
 * 2 lanes, and no exit
 */
type = 19;
}
else if ((vsegment->lateral_type == 0) &&
        (!vsegment->node_origin_entrance_exist) &&
        (vsegment->node_dest_exit_exist) &&
        (vsegment->node_dest_exit_type == 2) &&
        (vsegment->node_dest_exit_nblanes == 2))

```

```

{
/* no laterals, no entrance, and exit on the left-most lane
 * and 2 lanes
 */
type = 20;

}
else
{
type = 0;
}

if (type == 0)
return (0);

*nbsegments_affected = 1;

if (evaluate_only)
return (0);

/* fill the properties of this KRONOS segment */
segment->type = type;
unit = &(amp;segment->units[0]);
ksection = &(amp;unit->main);

if (error = TDReadSection (vsegment->idsection, &(ksection->TDsection)))
return (error);
ksection->dist_begin = vsegment->dist_begin;
ksection->dist_end = vsegment->dist_end;
ksection->nblaterals = 0;
/* by now */
ksection->metering_exist = FALSE;
TDKITTransferFlows (header, vsegment->flows, ksection->entrance_flows);

if (type == 2 || type == 4 || type == 5 || type == 7 || type == 8 ||
type == 10 || type == 17 || type == 19 || type == 21 ||
type == 23)
{
/* supply information for the entrance */
ksection = &(unit->entrance);

if (error = TDReadSection (vsegment->node_origin_entrance_idsection,
&(ksection->TDsection)))
return (error);
/* find section length */
if (error = TDGetSectionDim ((ksection->TDsection).trapezoid,
&section_length, &section_width))
return (error);
ksection->dist_begin = 0;
ksection->dist_end = section_length;
ksection->nblaterals = 0;
/* by now */
ksection->metering_exist = FALSE;
TDKITTransferFlows (header, vsegment->node_origin_entrance_flows,
ksection->entrance_flows);
}

if (type == 3 || type == 4 || type == 6 || type == 7 || type == 9 ||
type == 11 || type == 18 || type == 20 || type == 22 || type == 24)
{
/* supply information for the exit */
ksection = &(unit->exit);

if (error = TDReadSection (vsegment->node_dest_exit_idsection,

```

```

                                &(ksection->TDsection))
        return (error);
    /* find section length */
    if (error = TDGetSectionDim ((ksection->TDsection).trapezoid,
                                &section_length, &section_width))

        return (error);
    ksection->dist_begin = 0;
    ksection->dist_end = section_length;
    ksection->nblaterals = 0;
    /* by now */
    ksection->metering_exist = FALSE;
    TDKITransferFlows (header, vsegment->node_dest_exit_flows,
                      ksection->entrance_flows);
}

printf ("Now creating KRONOS segment type %d...\n", (int) segment->type);

/* translate this segment to KRONOS format */
if (error = TDKCreateSegment (segment))
    return (error);

return (0);
}
/*:..... end TDKVTreatment1V ..... */

ushort
TDKVTreatment2V_1 (TDKHeader * header, TDKSegment * segment, Bool evaluate_only,
                  ushort nbvsegments, TDKVSegment * vsegments,
                  Bool end_vsegments, ushort * nbsegments_affected,
                  Bool * cannot_decide_yet)
{
    /*:.....
    * Looks for a KRONOS segment that requires just two virtual segments,
    * from the beginning of the vsegments[] array.
    * Specifically, now it considers these types: 12-13.
    * If found, builds the appropriate call to TDKCreateSegment().
    *:.....
    */
    ushort error, type, nblaterals;
    float *flows, section_length, section_width;
    TDKVSegment *vsegment1, *vsegment2;
    TDKUnit *unit;
    TDKSection *ksection;
    TDLateral *pTDLaterals;

    *nbsegments_affected = 0;
    *cannot_decide_yet = FALSE;

    if (nbvsegments == 0)
        return (0);

    /* now consider the first segment */
    vsegment1 = &(vsegments[0]);

    if ((vsegment1->lateral_type == 1) &&
        (!vsegment1->node_origin_entrance_exist) &&
        (!vsegment1->node_dest_exit_exist))
    {
        /* right-entrance lateral, no entrance and no exit */
        type = 12;
    }
    else if ((vsegment1->lateral_type == 2) &&
             (!vsegment1->node_origin_entrance_exist) &&

```

```

        (!vsegment1->node_dest_exit_exist))
    {
        /* left-entrance lateral, no entrance and no exit */
        type = 13;
    }
    else
    {
        type = 0;
    }

    if (type == 0)
        return (0);
    if (nbvsegments < 2 && !end_vsegments)
    {
        *cannot_decide_yet = TRUE;
        return (0);
    }

    /* alright, from now it is clear that at least vsegment1 will be
     * considered
     */
    *nbsegments_affected = 1;

    if (nbvsegments >= 2)
    {
        vsegment2 = &(vsegments[1]);

        /* look whether vsegment2 also pertains to this segment */
        if ((type == 12 || type == 13) &&
            ((vsegment2->lateral_type == 0) &&
             (!vsegment2->node_origin_entrance_exist) &&
             (!vsegment2->node_dest_exit_exist)))
        {
            /* no laterals, no entrance and no exit */
            *nbsegments_affected = 2;
        }
        else
        {
            vsegment2 = NULL;
        }
    }
    else
    {
        vsegment2 = NULL;
    }

    if (evaluate_only)
        return (0);

    /* fill the properties of this KRONOS segment */
    segment->type = type;
    unit = &(segment->units[0]);
    ksection = &(unit->main);

    if (error = TDReadSection (vsegment1->idsection, &(ksection->TDsection)))
        return (error);
    ksection->dist_begin = vsegment1->dist_begin;
    ksection->dist_end = vsegment1->dist_end;
    if (vsegment2 != NULL)
    {
        ksection->dist_end += vsegment2->dist_end;
    }
    ksection->nblaterals = 1;
    if (error = TDReadSectionLaterals (vsegment1->idsection, &nblaterals,
                                     &pTDlaterals))

```

```

return (error);
/* to ease debugging */
memset ((char *) ksection->laterals, (char) 0, 2 * sizeof (TDLateral));

memcpy ((char *) &(ksection->laterals[0]), (char *) &(pTDLaterals[0]),
        sizeof (TDLateral));

/* by now */
ksection->metering_exist = FALSE;
TDKITTransferFlows (header, vsegment1->flows, ksection->entrance_flows);

printf ("Now creating KRONOS segment type %d...\n", (int) segment->type);

/* translate this segment to KRONOS format */
if (error = TDKCreateSegment (segment))
    return (error);

return (0);
}
/*..... end TDKVTreatment2V_1 .....*/

ushort
TDKVTreatment2V_2 (TDKHeader * header, TDKSegment * segment, Bool evaluate_only,
                  ushort nbvsegments, TDKVSegment * vsegments,
                  Bool end_vsegments, ushort * nbsegments_affected,
                  Bool * cannot_decide_yet)
{
/*.....
* Looks for a KRONOS segment that requires just two virtual segments,
* from the beginning of the vsegments[] array.
* Specifically, now it considers these types: 14-15.
* If found, builds the appropriate call to TDKCreateSegment().
.....*/
ushort error, type, nblaterals;
float *flows, section_length, section_width;
TDKVSegment *vsegment1, *vsegment2;
TDKUnit *unit;
TDKSection *ksection;
TDLateral *pTDLaterals;

*nbsegments_affected = 0;
*cannot_decide_yet = FALSE;

if (nbvsegments == 0)
    return (0);

/* now consider the first segment */
vsegment1 = &(vsegments[0]);

if ((vsegment1->lateral_type == 0) &&
    (!vsegment1->node_origin_entrance_exist) &&
    (!vsegment1->node_dest_exit_exist))
    {
    }
else
    {
    /* do not consider the first part as a 'simple straight' */
    vsegment1 = NULL;
    }

if (vsegment1 == NULL)
    {

```



```

    vsegment2 = &(vsegments[0]);
}
else if (nbvsegments < 2)
{
    if (!end_vsegments)
    {
        *cannot_decide_yet = TRUE;
    }
    return (0);
}
else
{
    vsegment2 = &(vsegments[1]);
}

if ((vsegment2->lateral_type == 4) &&
    (!vsegment2->node_origin_entrance_exist) &&
    (!vsegment2->node_dest_exit_exist))
{
    /* right-entrance lateral, no entrance and no exit */
    type = 14;
}
else if ((vsegment2->lateral_type == 3) &&
    (!vsegment2->node_origin_entrance_exist) &&
    (!vsegment2->node_dest_exit_exist))
{
    /* left-entrance lateral, no entrance and no exit */
    type = 15;
}
else
{
    type = 0;
}

if (type == 0)
    return (0);

if (vsegment1 == NULL)
{
    *nbsegments_affected = 1;
}
else
{
    *nbsegments_affected = 2;
}

if (evaluate_only)
    return (0);

/* fill the properties of this KRONOS segment */
segment->type = type;
unit = &(segment->units[0]);
ksection = &(unit->main);

if (error = TDReadSection (vsegment2->idsection, &(ksection->TDsection)))
    return (error);
ksection->dist_begin = vsegment2->dist_begin;
ksection->dist_end = vsegment2->dist_end;
if (vsegment1 != NULL)
{
    ksection->dist_end += vsegment1->dist_end;
}

```

```

}
ksection->nblaterals = 1;
if (error = TDRReadSectionLaterals (vsegment2->idsection, &nblaterals,
                                     &pTDlaterals))

    return (error);
/* to ease debugging */
memset ((char *) ksection->laterals, (char) 0, 2 * sizeof (TDLateral));

memcpy ((char *) &(ksection->laterals[0]), (char *) &(pTDlaterals[0]),
        sizeof (TDLateral));

/* by now */
ksection->metering_exist = FALSE;
TDKITransferFlows (header, vsegment2->flows, ksection->entrance_flows);

printf ("Now creating KRONOS segment type %d...\n", (int) segment->type);

/* translate this segment to KRONOS format */
if (error = TDKCreateSegment (segment))
    return (error);

return (0);
}
/*..... end TDKVTreatment2V_2 .....*/

```

```

ushort
TDKVTreatment3V (TDKHeader * header, TDKSegment * segment, Bool evaluate_only,
                ushort nbvsegments, TDKVSegment * vsegments,
                Bool end_vsegments, ushort * nbsegments_affected,
                Bool * cannot_decide_yet)
{
/*.....
* Looks for a KRONOS segment that requires just three virtual segments, *
* from the beginning of the vsegments[] array. *
* By now, this is just KRONOS segment 25. *
* If found, builds the appropriate call to TDKCreateSegment(). *
.....*/
ushort error, type, nblaterals;
float *flows, section_length, section_width;
TDKVSegment *vsegment1, *vsegment2, *vsegment3;
TDKUnit *unit1, *unit2, *unit3;
TDKSection *ksection;
TDLateral *pTDlaterals;

*nbsegments_affected = 0;
*cannot_decide_yet = FALSE;

if (nbvsegments == 0)
    return (0);

/* now consider the first segment */
vsegment1 = &(vsegments[0]);

if ((vsegment1->lateral_type == 3) &&
    (!vsegment1->node_origin_entrance_exist) &&
    (!vsegment1->node_dest_exit_exist))
{
    /* right-entrance lateral, no entrance and no exit */
    type = 25;
}
else
{
    return (0);
}
}

```

```

    }
}
if (nbvsegments < 2)
{
    if (!end_vsegments)
    {
        *cannot_decide_yet = TRUE;
    }
    return (0);
}

/* ok, now for the second segment */
vsegment2 = &(vsegments[1]);

if ((vsegment2->lateral_type == 0) &&
    (vsegment2->node_origin_entrance_exist) &&
    (vsegment2->node_origin_entrance_type == 1) &&
    (vsegment2->node_origin_entrance_nblanes == 1) &&
    (vsegment2->node_dest_exit_exist) &&
    (vsegment2->node_dest_exit_type == 1) &&
    (vsegment2->node_dest_exit_nblanes == 1))
{
    /* no laterals, entrance on the right-most lane and 1 lane,
    * and exit on the right-most lane and 1 lane
    */
    type = 25;
}
else
{
    return (0);
}
if (nbvsegments < 3)
{
    if (!end_vsegments)
    {
        *cannot_decide_yet = TRUE;
    }
    return (0);
}

/* ok, now for the third segment */
vsegment3 = &(vsegments[2]);

if ((vsegment3->lateral_type == 1) &&
    (!vsegment3->node_origin_entrance_exist) &&
    (!vsegment3->node_dest_exit_exist))
{
    /* right-entrance lateral, no entrance and no exit */
    type = 25;
}
else
{
    return (0);
}

/* -----allright, now it is already this type */
*nbsegments_affected = 3;

if (evaluate_only)
    return (0);

/* fill the properties of this KRONOS segment */
segment->type = type;
unit1 = &(segment->units[0]);
unit2 = &(segment->units[1]);
unit3 = &(segment->units[2]);

```

```

/* consider the first unit ----- */
ksection = &(amp;unit1->main);

if (error = TDRReadSection (vsegment1->idsection, &(ksection->TDsection)))
    return (error);
ksection->dist_begin = vsegment1->dist_begin;
ksection->dist_end = vsegment1->dist_end;
ksection->nblaterals = 1;
if (error = TDRReadSectionLaterals (vsegment1->idsection, &nblaterals,
                                   &pTDlaterals))

    return (error);
/* to ease debugging */
memset ((char *) ksection->laterals, (char) 0, 2 * sizeof (TDLateral));

memcpy ((char *) &(ksection->laterals[0]), (char *) &(pTDlaterals[0]),
        sizeof (TDLateral));

/* by now */
ksection->metering_exist = FALSE;
TDKITTransferFlows (header, vsegment1->flows, ksection->entrance_flows);

/* consider the second unit ----- */
ksection = &(amp;unit2->main);

if (error = TDRReadSection (vsegment2->idsection, &(ksection->TDsection)))
    return (error);
ksection->dist_begin = vsegment2->dist_begin;
ksection->dist_end = vsegment2->dist_end;
ksection->nblaterals = 0;

/* by now */
ksection->metering_exist = FALSE;
TDKITTransferFlows (header, vsegment2->flows, ksection->entrance_flows);

/* consider the third unit ----- */
ksection = &(amp;unit3->main);

if (error = TDRReadSection (vsegment3->idsection, &(ksection->TDsection)))
    return (error);
ksection->dist_begin = vsegment3->dist_begin;
ksection->dist_end = vsegment3->dist_end;
ksection->nblaterals = 1;
if (error = TDRReadSectionLaterals (vsegment3->idsection, &nblaterals,
                                   &pTDlaterals))

    return (error);
/* to ease debugging */
memset ((char *) ksection->laterals, (char) 0, 2 * sizeof (TDLateral));

memcpy ((char *) &(ksection->laterals[0]), (char *) &(pTDlaterals[0]),
        sizeof (TDLateral));

printf ("Now creating KRONOS segment type %d...\n", (int) segment->type);

/* translate this segment to KRONOS format */
if (error = TDKCreateSegment (segment))
    return (error);

return (0);
}
/*..... end TDKVTreatment3V ..... */

```

