INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the original text directly from the copy submitted. Thus, some dissertation copies are in typewriter face, while others may be from a computer printer.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyrighted material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each oversize page is available as one exposure on a standard 35 mm slide or as a 17" × 23" black and white photographic print for an additional charge.

Photographs included in the original manuscript have been reproduced xerographically in this copy. 35 mm slides or 6" × 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
A coordinated decentralized flow and routing control algorithm for an automated highway system

Sheu, Hsin-Teng, Ph.D.
The Ohio State University, 1987
PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark ✓.

1. Glossy photographs or pages
2. Colored illustrations, paper or print
3. Photographs with dark background
4. Illustrations are poor copy
5. Pages with black marks, not original copy ✓
6. Print shows through as there is text on both sides of page
7. Indistinct, broken or small print on several pages ✓
8. Print exceeds margin requirements
9. Tightly bound copy with print lost in spine
10. Computer printout pages with indistinct print
11. Page(s) lacking when material received, and not available from school or author.
12. Page(s) seem to be missing in numbering only as text follows.
13. Two pages numbered. Text follows.
14. Curling and wrinkled pages
15. Dissertation contains pages with print at a slant, filmed as received
16. Other

University Microfilms International
A COORDINATED DECENTRALIZED FLOW AND ROUTING
CONTROL ALGORITHM FOR AN AUTOMATED HIGHWAY
SYSTEM

A Dissertation
Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the
Graduate School of the Ohio State University

by

Hsin-Teng Sheu, B.S.E.E., M.S.E.E.

* * * * *

The Ohio State University
1987

Dissertation Committee:
Prof. Robert E. Fenton
Assoc. Prof. Ümit Özgüner
Assist. Prof. Stephen Yurkovich

Approved by:

Robert E. Fenton
Adviser
Department of Electrical Engineering
ACKNOWLEDGEMENTS

The author wishes to express his sincere appreciation to his advisor Prof. Robert E. Fenton for his constant and invaluable guidance and inspiration. He also wishes to appreciate the helpful comment and suggestion by Prof. Ümit Özgüner and Prof. Stephen Yurkovich as well as the essential support provided by Prof. H.C. Ko, The Chairman of The Department of Electrical Engineering at The Ohio State University, National Science Council of Republic of China at Taiwan, Prof. Yen-Ping Shih, The President of National Taiwan Institute of Technology, and Prof. Shen-Nian Yeh, The Chairman of The Department of Electrical Engineering at NTIT. His special thanks go to the faculty of the Department of Electrical Engineering, the Department of Computer and Information Science, and the Department of Mathematics at The Ohio State University for their helpful instruction. He also wishes to thank his parents, his wife, and his family members for their everpresent support.
VITA

October 20, 1949 .........................Born—Kaohsiung, Taiwan, R.O.C.

1971 ...........................................B.S.E.E., National Chiao-Tung University, Taiwan, R.O.C.

1972-1974 ....................................Maintenance Engineer, General Instrument Microelectronics at Taiwan

1975 ...........................................M.S., University of Cincinnati, Cincinnati, Ohio

1976-1977 ....................................Research Engineer, Orient Semiconductor, Taiwan

1978-1979 ....................................Research Engineer, Industrial Technology Research Institute, Taiwan

1980-1981 ....................................Manager, Strong Electronic Company, Taiwan

1981-Present .................................Instructor, National Taiwan Institute of Technology, Taiwan

FIELDS OF STUDY

Major Field: Control Systems (Profs. R.E. Fenton and Ü. Özuguner)

Minor Fields: Communication Systems (Prof. F.D. Garber),
Computer and Information Science (Prof. Ming-Tsang Liu),
Mathematics (Prof. T. Schiek)
TABLE OF CONTENTS

ACKNOWLEDGEMENTS ii
VITA iii
LIST OF FIGURES vi

I. INTRODUCTION 1
1.1 Routing in Transportation Networks 2
1.2 Routing in Communication Networks 6
1.3 Comparison of Communication and Transportation Networks 8
1.4 Proposed Approach 10
1.5 Outline 11

II. COORDINATED ROUTING CONTROL FOR A SINGLE-DESTINATION DETERMINISTIC NETWORK 12
2.1 Introduction 12
2.2 The Model of A Typical Node 13
2.3 Single-Destination Routing 17
2.4 Single-Destination Routing — Determination of Input and Output Traffic 22
2.4.1 Optimal Solutions 24
2.4.2 Optimal Controls Do Not Exist 50

iv
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A General Network Model</td>
</tr>
<tr>
<td>2</td>
<td>A Node Model</td>
</tr>
<tr>
<td>3</td>
<td>Coordination Between the Local and the Upstream Nodes</td>
</tr>
<tr>
<td>4</td>
<td>Physical Geometry of Single Destination Routing</td>
</tr>
<tr>
<td>5</td>
<td>Single Destination Routing</td>
</tr>
<tr>
<td>6</td>
<td>Routing Model</td>
</tr>
<tr>
<td>7</td>
<td>Domain associated with Unsaturated Controls ($t_f$ free)</td>
</tr>
<tr>
<td>8</td>
<td>Domain associated with Saturated $u_r$, $(1 + a_2) C_f \geq a_2 r_D + C_r$</td>
</tr>
<tr>
<td>9</td>
<td>Domain associated with Saturated $u_r$, $(1 + a_2) C_f &lt; a_2 r_D + C_r$</td>
</tr>
<tr>
<td>10</td>
<td>Domain associated with Saturated $u_f$ ($t_f$ free)</td>
</tr>
<tr>
<td>11</td>
<td>Domain associated with Saturated Input and Output Traffic</td>
</tr>
<tr>
<td>12</td>
<td>Possible trajectories in the Region Associated with Saturated Controls</td>
</tr>
<tr>
<td>13</td>
<td>Domains for Controls — case a</td>
</tr>
<tr>
<td>14</td>
<td>Domains for Controls — case b</td>
</tr>
<tr>
<td>15</td>
<td>Domains for Controls — case c</td>
</tr>
<tr>
<td>16</td>
<td>Domain Associated with Unsaturated Controls, $t_f$ fixed</td>
</tr>
<tr>
<td>17</td>
<td>Domain associated with Saturated $u_r$, $t_f$ fixed</td>
</tr>
<tr>
<td>18</td>
<td>Possible trajectories in the Region Associated with Saturated $u_r$, $t_f$ fixed</td>
</tr>
<tr>
<td>Page</td>
<td>Section</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>19</td>
<td>Network for Simulation Studies</td>
</tr>
<tr>
<td>20</td>
<td>Traffic under constant, moderate demand</td>
</tr>
<tr>
<td>21</td>
<td>Queues under constant, heavy demand</td>
</tr>
<tr>
<td>22</td>
<td>Traffic under constant, heavy demand</td>
</tr>
<tr>
<td>23</td>
<td>Travel time as a function of demand</td>
</tr>
<tr>
<td>24</td>
<td>Queues under constant, moderate demand with initial queues</td>
</tr>
<tr>
<td>25</td>
<td>Traffic under constant, moderate demand with initial queues</td>
</tr>
<tr>
<td>26</td>
<td>Traffic under sinusoidally changing, moderate demand</td>
</tr>
<tr>
<td>27</td>
<td>Queues under sinusoidally changing, heavy demand</td>
</tr>
<tr>
<td>28</td>
<td>Traffic under sinusoidally changing, heavy demand</td>
</tr>
<tr>
<td>29</td>
<td>Queues under moderate demand, with exit congestion</td>
</tr>
<tr>
<td>30</td>
<td>Traffic under moderate demand, with exit congestion</td>
</tr>
<tr>
<td>31</td>
<td>Traffic under 3 moderate demands</td>
</tr>
<tr>
<td>32</td>
<td>Queues under 3 heavy demands</td>
</tr>
<tr>
<td>33</td>
<td>Traffic under 3 heavy demands</td>
</tr>
<tr>
<td>34</td>
<td>A highway system in a typical in-land city</td>
</tr>
<tr>
<td>35</td>
<td>Destinations associated with which the downstream links are inadmissible</td>
</tr>
<tr>
<td>36</td>
<td>Inadmissible destinations on each link at Node 1</td>
</tr>
<tr>
<td>37</td>
<td>Traffic under moderate demand — nodes 1, 2, 3, and 4</td>
</tr>
<tr>
<td>38</td>
<td>Traffic under moderate demand — nodes 5, 6, 7, and 8</td>
</tr>
<tr>
<td>39</td>
<td>Queues at nodes 1 through 4 under heavy demand levels</td>
</tr>
<tr>
<td>40</td>
<td>Queues at nodes 5 through 8 under heavy demand levels</td>
</tr>
<tr>
<td>41</td>
<td>Traffic at nodes 1 through 4 under heavy demand levels</td>
</tr>
<tr>
<td>42</td>
<td>Traffic at nodes 5 through 8 under heavy demand levels</td>
</tr>
</tbody>
</table>
43 Queues at nodes 1 to 4 under moderate demands, \( q_1(0) = 50 \) . . . 102
44 Queues at nodes 5 to 8 under moderate demands, \( q_1(0) = 50 \) . . . 103
45 Traffic at nodes 1 to 4 under moderate demands, \( q_1(0) = 50 \) . . . 104
46 Traffic at nodes 5 to 8 under moderate demands, \( q_1(0) = 50 \) . . . 105
47 Queues at nodes 1 to 4 under time-varying demands . . . . . . . 107
48 Queues at nodes 5 to 8 under time-varying demands . . . . . . . 108
49 Traffic at nodes 1 to 4 under time-varying demands . . . . . . . 109
50 Traffic at nodes 5 to 8 under time-varying demands . . . . . . . 110
51 Queues at node 1 — 4 under stochastic, heavy demands . . . . . 113
52 Queues at node 5 — 8 under stochastic, heavy demands . . . . . 114
53 Traffic at node 1 — 4 under stochastic, heavy demands . . . . . 115
54 Traffic at node 5 — 8 under stochastic, heavy demands . . . . . 116
55 Queues at node 1 — 4, limited egress-capacity case . . . . . . . 118
56 Queues at node 5 — 8, limited egress-capacity case . . . . . . . 119
57 Traffic at node 1 — 4, limited egress-capacity case . . . . . . . 120
58 Traffic at node 5 — 8, limited egress-capacity case . . . . . . . 121
59 The hierarchy of the program modules . . . . . . . . . . . . . . . . 140
60 The flowchart of NETWORK . . . . . . . . . . . . . . . . . . . . . . 141
61 The flowchart of NOEDYN . . . . . . . . . . . . . . . . . . . . . . 142
62 The flowchart of ALLOCATION . . . . . . . . . . . . . . . . . . . . 143
CHAPTER I
INTRODUCTION

The need for good transportation has existed since the early days of mankind. In modern times, this need has become essential as the economic health of a nation depends, in large measure, upon the efficacy of its transportation capabilities. One especially critical aspect is the roadway system which, in this country, is frequently characterized by overcrowding, an excessive number of accidents, and negative effects on the environment. At present these problems are caused by some 128 million motor vehicles [1]; since it is anticipated that this number will double by the year 2004, this situation can only become worse.

Conventional approaches such as the building of additional roadways, the increased use of mass transit, and improved control of freeway traffic will, at best, provide only a partial solution. It thus becomes essential to examine promising new and innovative approaches. One of these is an automated highway system (AHS) which, in contrast to today’s highway system, would utilize a computerized vehicle and roadway system to achieve both individual vehicle control and network control. The former would involve longitudinal and lateral control of individual vehicles [2]–[11] whereas the latter would include entry control, merging control, routing, scheduling, and the system’s response to failures and emergencies. Since computers would be used to monitor traffic conditions and issue commands accordingly, high flow rates could be achieved, — possibly to the limit specified by the permitted
intervehicular spacing. In particular, if a small intervehicular spacing (e.g., 40 feet at a traffic speed of 55 miles per hour) could be achieved, highway capacity could be substantially increased without compromising safety.

Among the control functions involved in an AHS, route guidance would be especially important as it would include how vehicles are allocated on each link. If it were effectively employed, it could result in a more efficient utilization of the roadway facilities. The benefits could include a reduction in both mean travel time and its variance, a lessening of congestion at peak hours, and a means of handling certain anomalous or emergency situations. For these reasons, route guidance in an AHS will be the focus of this dissertation.

1.1 Routing in Transportation Networks

Many recent works have been focused on the control and performance evaluation of transportation networks but few of these were on routing [12]–[17]. One reason is that in the past, cheap, intelligent control devices, which would be essential for routing, were not available; instead, control functions in transportation networks were usually achieved by using hardwired circuits. As a result, most activities were focused on achieving control in a limited area — generally signal control at an intersection or intersections in an urban environment [18]–[25]. Therefore, before formulating solutions to routing problems, it would be useful to review such studies to gain insight.

In general, a microscopic approach to traffic control would account for the state of every vehicle; this approach can involve excessive computer storage and computational requirements and thus appears impractical. In a macroscopic approach, traffic is treated in the aggregate, frequently in terms of fluid flow, and the resulting traffic controllers are simpler and less costly to implement. For ex-
ample, effective signalization can be achieved using aggregated parameters such as the average number of vehicles in a platoon and/or the manner in which this estimate changes over time [18]. In [19]–[20], continuous state equations were used to represent traffic dynamics at a single intersection and traffic was controlled by minimizing a linear functional of queue length. In [21], Gazis formulated the dynamics of the queue lengths in a network of intersections as a parametric linear programming problem and obtained the solution by compromising between two competing input traffic streams.

A formulation that used a delayed, discrete-time, state-transition model was also suggested [22]–[24]. By employing time decomposition, as well as component decomposition, a two-level optimization scheme for intersection control resulted. A discrete-time model without delay was also proposed [25] for an urban intersection. Here, different traffic conditions, or modes, were associated with different objective functions: minimum queue length for the heavy traffic mode, maximum average flow for the congested traffic mode, and minimum driver discomfort for the stopped traffic mode.

Three fundamental works on the purpose of routing were found in [12,13]: a rest-stop survey in Houston, a commuter survey in Dallas, and a survey using a stated-preference approach (one that involves rating or ranking a set of alternatives) in the Netherlands. The first two of these pertained to motorists whereas the third pertained to bicyclists. Although slight variations in the perception of the relative importance of routing criteria were noted, convenience and travel time were found to be the most important. In the rest-stop survey, in which 76% of the respondents were unfamiliar with the area, the most direct or shortest route was also considered to be important, and in the commuter survey, fewer stops had a substantial effect on the route-choice decision. Further, the first two surveys
revealed that less congestion and good traffic flow sometimes might result in the choice of a particular route over the others. On the other hand, the last survey showed that, in the order of their relative importance, road-surface quality, traffic level, and facility type were the most important factors in the route-choice decision process.

The potential advantages of routing were recognized early in the mid 60's and a Driver Aid Information and Routing system (DAIR) was developed by General Motors Corporation [26]. In the late 60's, the Federal Highway Administration sponsored studies of an Electronic Route Guidance System (ERGS) [27]. Subsequently, after the U.S. Congress refused to continue this program, Japanese, British, and German investigators developed the Comprehensive Automobile Traffic Control System (CACS), Advance WARning Equipment (AWARE), and Automotive Road Information Traffic Control System (ARI), respectively [26,28]. The basic approach in these systems is to use a route table, preset according to historical data or frequently updated based on traffic conditions, to guide vehicles. In all these efforts, the focus was on route guidance as an aid to the driver, and the goal was improved flow on the present roadway system.

Recently, Chu [14], Sarachik and Özgüner [15], Sarachik [16], and Ng [17] used a macroscopic approach to formulate the routing problem for the constant traffic-demand case. In the first, continuous as well as discrete models were used and it was found that the optimal solution for the continuous case converged to that obtained in the discrete case under stochastic input conditions. In the second and the third studies, a decentralized routing problem was formulated by using a continuous state equation for each node and both routing costs and queue length were incorporated in the objective functions. Like most constrained optimization problems, this resulted in a bang-bang type solution. In the third study, a discrete
model was used and a quadratic cost function was considered. Component decom­
position, or decentralization, as well as time decomposition was employed, and a
two-level hierarchical control problem resulted. A drawback is that the feedback
solution is implicit and an iterative process is required.

Two points should be noted from this brief overview: a macroscopic approach
was employed in all of these studies; and the focus was on simple highway geome­
tries with simple input functions. General networks with stochastic traffic inputs
have not been considered nor have general principles for dynamic routing been
specified. In view of the potential importance of this subject, it appears to be a
profitable area for investigation.

In the context of an AHS, a routing strategy would probably be different from
that for a normal highway. Firstly, choices related to individual preference, such
as convenience and accessibility might be relatively unimportant for an AHS since
the goal would be optimum system performance. Secondly, it seems reasonable to
assume that adequate space for maneuvering, sufficient storage space for queuing,
and reliable communication links would be available. In formulating the problem,
routing criteria such as short trip, less congestion, fewer stops, and good traffic
flow could be accounted for by specifying an objective function which included
queue length, link travel-time, and the error in tracking the input demand. Also, a
road-surface quality measure could be implicitly contained in a link capacity term.
Appropriate controls would be determined by minimizing this function.

Several studies have been concentrated on network control in an AHS [29] –
[33]; however, all of these have dealt with static, or fixed, routing. Further, since
relatively little effort has been focused on routing in transportation networks, rout­
ing studies in other areas might provide some insights. Routing problems arise in a
variety of situations, including those dealing with water resources, printed circuits,
integrated circuits, pipeline layout, electrical power distribution, and communication networks. A number of studies have been conducted relative to each of these; however, only those associated with communication systems appear to be relevant to the highway routing problem.

1.2 Routing in Communication Networks

Most routing studies for communication networks have involved message-switched and virtual-circuit, packet-switched networks [34]–[52]. The routing problem, which is of a quasi-static nature in such networks, involves the determination of an optimal physical or logical connection before a message or a collection of packets is sent [53]. Several models of this process have been proposed, including: finite-state transition models [34,35],[38,43,63], queueing models [39,40,41,42,36], and simulation models [46]–[48]. In addition, some researchers have formulated approaches which are analogous to problem-solving techniques in other areas [37,44].

The finite-state model appears to be most applicable to highway traffic routing. Also, it is the most interesting because of its mathematical rigor and the optimization technique employed. In this formulation, the queue length is taken as a state variable, the routed traffic is treated as the control variable and the input traffic is considered to arrive at a fixed rate, (i.e., an estimated mean arrival rate is assumed to be fixed). The model is constrained to comply with the conservation of flow and a cost function, which is a functional of queue length, is formulated. Consequently, an explicit feedback solution is possible.

In the queueing model formulation, an M/M/1 queue\(^1\) is assumed and a formula for average time delay in the network is derived, with flow as an independent

\(^1\)A single server queue is called an M/M/1 queue if the arrival and service rates are exponentially distributed.
variable [54]. The task is to find the optimal flow which minimizes the average time delay. Several methods for achieving this have been proposed: a flow deviation method [39], an extremal flow method [40], a gradient projection method [41], an iterative approach [42], and a best deterministic rule [36]. In all cases, the solution is not in an explicit feedback form.

The simulation approaches involve various heuristic algorithms, such as a minimal spanning tree (MST) algorithm [55], a minimal destination cost (MINDC) algorithm, and nearby heuristic (NH) and nearest neighbor (NN) heuristic algorithms [46]. Since these are developed heuristically, optimality cannot be guaranteed.

The problem-solving techniques employed for network routing include approaches involving diffusion [37], and relaxation [44]. In the former, routing is modeled as a diffusion process and a partial differential equation is formulated. The nature of this formulation is that neither the existence nor the uniqueness are assured; further, only limited solution methods are available. The latter approach uses an analogy between electrical networks to formulate network parameters in communication in terms of those in electrical networks, such as traffic demand as current source, link flow as branch current, routing link as nonlinear resistor, and destination node(s) as ground(s), respectively, and analyze the network by a relaxation technique, i.e., solving for the optimal flow iteratively among neighboring nodes [56]. Although this approach has been successful in off-line computations for the single-destination case, it is probably not practical for the on-line operation of a multi-destination network since the computation involved is significant, and the solution may not be unique.

Two control structures, centralized [34]–[38] and decentralized [42]–[63], can be employed in a corresponding implementation. In the former, a central node
would be used to implement routing by collecting network information, making
decisions based on optimizing the cost function, and then sending control com-
mands to the nodes in the network. This approach seems straight forward and
simple to implement; however, it is neither. The difficulties involve reliability and
dimensionality. A failure at either the central node or any link to that node could
affect the whole network. Furthermore, for a network with N nodes, the dimension
of the problem, relative to the computation load, will be N(N-1) thus making this
load unacceptable for large N.

In the decentralized approach, each node would collect information from neigh-
boring nodes and the decision-making and control would be accomplished locally.
Since decisions would be made in a distributed way and the information used would
be aggregated, the network operation would be generally more reliable; however,
the solution would be suboptimal.

1.3 Comparison of Communication and Transportation Networks

Although many of the approaches proposed for communication networks ap-
pear applicable to transportation networks, they cannot be used without modifi-
cation due to the following intrinsic differences:

Delay The transport delay between an upstream and downstream node is negli-
gible in a communication network but not in a transportation network;

Ordering The order of departure of messages can be arbitrarily specified in com-
munication networks but not in transportation networks (It could be speci-
fied in the latter if a number of departure bays were available; however, in
general, this would be impractical);
**Duplicatability** Messages can be duplicated and bifurcated for routing to several destinations. Clearly this cannot be done for vehicles;

**Turnaround** Computation time for route selection generally must be done in milliseconds for communication networks whereas for transportation networks this can be relaxed to seconds;

**Update frequency** Route selection may not be changed frequently in communication networks as this would require a considerable overhead of channel capacity. In a transportation network, however, a dedicated communication network could be used to pass information between neighboring nodes to achieve dynamic routing; and

**Size** The message size may vary in communication networks (packet-switched or message switched) whereas vehicles are fixed entities.

Such differences clearly have some impact on a route selection algorithm for a transportation network.

In designing a transportation network, the link travel time must be included and the corresponding delays incorporated in the dynamic equations. The analysis thus becomes more difficult than in the communication case. The sequential nature of vehicle ordering, in contrast to the combinational nature of message ordering, makes multi-destination routing more complicated since the entering traffic can no longer be expressed by simply summing that from upstream links. Since messages can be duplicated, message delays in the whole network can be reduced, although the required tasks may be computationally difficult [46,57]. The longer turn-around time associated with transportation networks may allow more choices among various routing strategies. Also with frequent information updates, the
network status as well as the traffic conditions can be closely monitored so that the performance over time may be closer to optimum. The variable-message size in communication networks results in much computation in solving for message rates, or flow, whereas fewer computations should be required for transportation networks.

In essence, while many similarities exist between the two network types, results obtained for one case cannot be applied without modification to the other.

1.4 Proposed Approach

In virtually all of the reported works, whether focused on communication or transportation networks, it was assumed that the traffic entering each node was uncontrolled. In a network, this could result in congestion and excessive delay at downstream nodes. To avoid these problems, it is necessary to control the flow entering each node. Thus, flow control and routing should be considered simultaneously to achieve satisfactory network performance [47].

By incorporating flow control in a system for dynamic routing, it is hypothesized that traffic flow should be smoother, serious congestion less probable, and the network safer. The implementation would involve coordination between neighboring upstream and downstream nodes. Such coordination can generally be achieved in either a centralized or a decentralized framework. In the former, the coordination would be implicit since interactions among nodes would be contained in the system dynamics. In the latter, communication of relevant information, such as measured queue lengths and control commands, would be required. Here, for reasons of reduced computational complexity and enhanced network reliability as previously discussed, the latter will be employed.

In a decentralized framework, the coordinated routing problem will be divided
into two parts: routing and flow control. The former will involve the determination of the locally optimal routing and aggregated flow control whereas the latter will include the prediction of traffic demand and distribution of allowed traffic among upstream links. The formulation of the dynamics will thus involve an ordinary, non-delayed differential equation containing an aggregated flow control term and the objective function would include a tracking error, which is the difference between the input traffic demand and the actual traffic flow, as well as queue lengths and routing costs. After the optimal aggregated flow control is determined, the allowed traffic on each upstream link can be allocated according to their requested traffic.

1.5 Outline

This dissertation is organized as follows: In Chapters 2 and 3, the problem is formulated, locally optimal solutions are determined, and a simulation study involving a single-destination network is presented. In Chapter 4, the routing control for a multi-destination network and a corresponding the simulation study, which involves a prototype AHS configuration, is presented. The results obtained are compared with those from an earlier study where only fixed routing was employed [33]. Finally, summary and a detailed discussion of the results is contained in Chapter 5, together with suggestions for future studies.
CHAPTER II
COORDINATED ROUTING CONTROL FOR A
SINGLE-DESTINATION DETERMINISTIC NETWORK

2.1 Introduction

Consider the routing problem for a general automated network with m inputs: $I_1, ..., I_m$, and n outputs: $O_1, ..., O_n$, as shown in Fig. 1. Usually, such a network would consist of a number of interconnected nodes; some of these would be source nodes, i.e., nodes from which vehicles start their journey, some would be destination nodes, i.e., nodes at which vehicles leave the network, and some would be internal nodes where traffic merging and/or diverging occurs. Each source node may have several input links (or inputs), each destination node may have several output links (or outputs), and each merging or diverging node may be connected to several links. Further, vehicles departing from a source node may have a variety of different destinations, and vehicles arriving at a destination node may have come from various source nodes. The routing problem in such a network involves choosing a route for a vehicle so that certain criteria are satisfied — ideally in an optimal manner. Either a decentralized or centralized approach may be employed. The disadvantage of the former is that the solution might be only suboptimal; however, since excessive communications, extensive computations and relatively low reliability could result from using the latter, a decentralized scheme will be employed.
In a decentralized framework, a best route for each vehicle would be determined based on the information from its neighboring nodes. Basically, the decision-making process at each node would be similar and it suffices to consider only a typical node.

2.2 The Model of A Typical Node

In the node model shown in Fig. 2, \( N_o \) is such a node, \( N_{S_1}, \ldots, N_{S_k} \) are upstream neighboring nodes, and \( N_{D_1}, \ldots, N_{D_n} \) are destination nodes. Note the existence of several routes to each destination.

In general, this model can be described by either a continuous or a discrete representation. Since traffic has a discrete nature, the latter would be a natural choice. Usually, the solution method associated with this formulation involves time decomposition and dynamic programming techniques. The discrete version of Pontryagin’s minimum principle, or the Kuhn-Tucker condition, is employed and the resulting feedback solution is implicit, the dimensionality is high [58,59],
and an iterative process might be necessary [17], [22] – [24].

In the continuous model, the solution method involves minimizing an objective function subject to differential-equation constraints. Calculus of variations and other mathematical methods are available for solving the problem, which usually results in an explicit feedback solution. Further, although the continuous model is not a natural representation, it may be a good approximation to discrete systems since, according to Chu [14], the results obtained by applying two optimal strategies for a discrete model of a traffic situation converge to those from corresponding continuous models. Because the mathematics associated with the continuous model appear to be more tractable and such a formulation could be a good approximation, especially in congested situations when information updating would be done frequently, it will be employed here.

In general, the estimated routing costs (or delays) on the downstream links would be obtained at \( N_0 \), and vehicles would be released by compromising between these costs, the queue at that node, and the anticipated upstream demand. The queue would result from surging demand, limited capacities on the downstream links, and the routing (by several upstream controllers) of traffic through \( N_0 \) on the basis of little present congestion at that node thereby causing future congestion. However, if appropriate coordination between upstream and downstream nodes were provided, the former could dispatch only that quantity of traffic which should not cause congestion.

The functions performed at \( N_0 \) can be divided into three parts:

1. Estimation of requesting traffic demand from upstream nodes and the queue lengths at both those nodes and \( N_0 \);

2. Determination of the aggregated input traffic acceptable at \( N_0 \) and the spec-
ification of the total flow rate on downstream links; and

3. Allocation of acceptable traffic rates from upstream links and output traffic on downstream links.

Note that transport delays are only considered explicitly in the first and the third parts.

It is hypothesized that these functions can be efficiently performed by the protocol specified in Fig. 3. The quantities employed here are defined as follows:

- $d_i$ is the transport delay, in minutes, associated with the upstream link $i$, ($d_i \leq \ldots \leq d_k$);
- $r_{D_j,S_i}(t)$ is the optimal desired departure rate from $N_{S_i}$ at time $t$ (heading for destination $D_j$ by way of $N_o$) as computed at $N_{S_i}$. (In general, the vehicles at $N_{S_i}$ will be dispatched both to $N_o$ and other downstream nodes.);
- $r_{D_j,i}(t) = r_{D_j,S_i}(t - d_i)$ is the arrival rate at $N_o$ from $N_{S_i}$ at time $t$ with destination $D_j$ if $N_o$ accepts the desired number of vehicles from $N_{S_i}$;
- $r_{D_j}(t) = \sum_{i=1}^{k} r_{D_j,i}(t)$ is the aggregate arrival rate or demand with destination $D_j$ which, if serviced, would arrive at $N_o$ at time $t$;
- $\bar{r}_{D_j,i}(t)$ is the predicted $r_{D_j,i}(t)$;
Figure 3: Coordination Between the Local and the Upstream Nodes
$\bar{r}_{D_i}(t)$ is the predicted $r_{D_i}(t)$;

$u_{D_i,j}(t-d_i)$ is the departure rate from $N_{S_i}$ that is acceptable at $N_o$ at time $t$, i.e.,

the flow control on the upstream link $i$;

$u_{D_i,j}(t) = \sum_{i=1}^{k} u_{D_i,j}(t - d_i)$ is the aggregate departure rate from all upstream links

acceptable at $N_o$, and corresponds to the acceptable arrival rate at $N_o$ at time

$t$;

$u_{D_i,r}(t)$ is the routing control, or desired flow rate, on the downstream link $i$ at $N_o$ at

time $t$, $i=1, ..., I$;

$u_{D_i,r}(t) = \sum_{i=1}^{I} u_{D_i,r}(t)$ is the aggregated output traffic rate at $N_o$ at time $t$.

Here all of the rates are in vehicles/min.

Note that the controller at $N_o$ is required to communicate only with its nearest
upstream and downstream neighboring nodes. Further, this protocol is intended
only for normal, nonemergency operations. In an emergency situation, such as link
blockage due to an accident, communication with additional control entities would
probably be necessary.

### 2.3 Single-Destination Routing

In view of the complexity of the multi-destination routing problem, it may be
useful to model the single-destination case first in order to gain insight. Consider
the physical geometry of the node shown in Fig. 4, where $N_{S_1}, ..., N_{S_k}$ are up-
stream nodes, $N_D$ is the destination node, and the rectangles represent the queues
formed on merging links. These queues represent not only stopped vehicles but
also maneuvering ones, especially those of the slow down-speed up variety since
these would result in an increase in travel time which can be viewed as equivalent
to delay by queuing.

Since several queues are associated with each node, they may be aggregated
as a single one and put in front of each node to simplify the representation [15].
Further, in the decentralized framework, incoming traffic to the upstream nodes
Figure 4: Physical Geometry of Single Destination Routing

can be considered collectively since, it is that traffic which affects the control
decision at $N_0$. Consequently, a diagram as shown in Fig. 5 results. For single-
destination routing, this is justified since the ordering of incoming traffic is not
important. For the multi-destination case, however, this is not true as will be
discussed later.

A unique approach, which is developed by this author, is employed to model
the operations at a typical node $N_0$. Two queues are involved, one at $N_0$ the
other at upstream nodes. The former, represented as $q_1(t)$, is due to the difference
between the input and output rates. More precisely, it is a function of the input
and output traffic conditions at that node, the downstream link capacities, and the
control law employed. The latter arises because of the difference between the input
demand and the acceptable traffic at $N_0$. For example, a queue $q_{S_i}(t - d_i)$ could
form at $N_{S_i}$ at time $t - d_i$ because of surging traffic demand and the projected flow
control applied at $N_0$ at time $t$. Conceptually, all such queues can be considered
collectively at $N_o$ and represented as $q_2(t) = \sum_{i=1}^{k} q_{S_i}(t - d_i)$, and the model for single-destination routing shown in Fig. 6 results. Note that $q_2(t)$ is placed behind $N_o$ to reflect the fact that it is associated with the flow control applied from that node.

In traffic control queue length is usually considered to be a measure of congestion, and it is convenient to include such measures in a model of node operations. Here, both $u_f(t) - u_r(t)$ and $r_D(t) - u_f(t)$ are included\(^2\), with the former being a measure of the change of $q_1(t)$ at $N_o$, and the latter a measure of the change of $q_2(t)$, i.e., the fictitious queue upstream from $N_o$. The corresponding queue

\(^1\)If released at times $t - d_1, \ldots, t - d_k$, the vehicles in this "queue" would arrive at $N_o$ at time $t$; i.e., this equation is referenced to $N_o$ at time $t$.

\(^2\)For the single-destination routing problem, the notation has been simplified so that the destination $D$ does not show explicitly except in $r_D$. 

Figure 5: Single Destination Routing.
dynamics can be described by:

$$\dot{q}_1(t) = u_f(t) - u_r(t)$$  \hspace{1cm} (2.1a)

and

$$\dot{q}_2(t) = r_D(t) - u_f(t).$$  \hspace{1cm} (2.1b)

Note that the first measure would accurately reflect the change in $q_1$ if $u_r(t)$ were an accurate measure of the downstream flow from $N_o$. Normally, this flow is specified by controllers downstream from $N_o$, and this flow is not necessarily $u_r(t)$; however, $u_r(t)$ should be a good approximation. The second measure deals with the effectiveness of $N_o$ in servicing the optimal desired flow from upstream nodes.

The coordination between $N_o$ and the downstream nodes, in terms of (2.1a) and (2.1b) should result in a near-optimal balance between congestion and smooth traffic flow; however, it might not be able to sufficiently limit traffic from entering $N_o$ when either $N_o$ was congested or $r_D$ were larger than the aggregated upstream or downstream capacities, $C_f$ and $C_r$. In such situations, the coordination could
be facilitated if the queue at $N_0$ were viewed as comprised of two parts: one part, $u_f - u_r$ per equation (2.1a), results from the control at $N_0$ and the other, $q'_1$, from the control exercised by downstream nodes; i.e., $q'_1$ is the difference between $u_r$ calculated at $N_0$ and the flow acceptable to downstream nodes. One could collect $q'_1$ into $q_1$ at the end of each control interval so that each control law would start with $q'_1 = 0$ and $q_1$ would reflect the actual queue size; however, this will not be done, since, if $r_D < C_f$ and $r_D < C_r$, $q_1$ and $q_2$ can be cleared and it would not be necessary to do so. Further, $q'_1 = 0$ means that the $q_2$'s associated with the neighboring downstream nodes would become zero; i.e., those nodes would lose the measure of their effectiveness in servicing the upstream demand. Moreover, since control intervals would not be synchronized among nodes (as will be explained in the next paragraph), this clearing would result in a step change at downstream nodes and a corresponding increase in the complexity of computations. Therefore, $q'_1$ will not be reset unless either $r_D$ were larger than $C_f$ or $C_r$ for a sufficient time or $q'_1$ were greater than a preselected threshold $q'_1p$. Upon resetting, $q_1$ would become larger and the acceptable input traffic would be restricted. Successive application of such a rearrangement on the queues at upstream nodes would therefore inhibit excessive traffic from entering the network and prevent further internal congestion.

It is necessary to employ updating so that traffic control can be adapted to changing conditions. Such updating could be achieved synchronously or asynchronously. In the former, the $r_D$ associated with each network node would be updated, new control laws would be selected and then applied simultaneously. In the latter, these functions would still be performed but the laws would be applied asynchronously, i.e., the control intervals would not be synchronized among nodes. Such an arrangement should reduce the complexity of operations with respect to the synchronous case since each control law would be determined locally.
Therefore, this will be employed.

2.4 Single-Destination Routing — Determination of Input and Output Traffic

In general, the performance of an automated network can be judged by two metrics: congestion and smoothness. The former, which relates to delay and thus to trip time, can be measured by the queue lengths and the latter by the difference between the traffic demand and the approximate traffic rate $u_r(t)$.

An approach to node control involves minimizing an objective function subject to the dynamic constraints, equations (2.1a) and (2.1b), and the following positivity constraints:

\begin{align*}
q_1(t) & \geq 0 & (2.2a) \\
q_2(t) & \geq 0 & (2.2b) \\
r_D(t) & \geq 0 & (2.2c) \\
C_f & \geq u_f(t) & \geq 0 & (2.2d) \\
C_r & \geq u_r(t) & \geq 0 & (2.2e)
\end{align*}

\forall t.

Three types of objective functions were considered for use here: linear, quadratic with $u_r$ but not $u_f$, and quadratic in both $u_r$ and $u_f$. The first involved a functional of a weighted sum of queue lengths and the absolute difference between $u_r$ and $r_D$. The corresponding controls, i.e., the allowed input and output traffic rates are of a bang-bang type. This means that either no traffic or maximum traffic is allowed to enter or leave $N_o$ and vehicles on the links into or out of $N_o$ could stop.

\[\text{This is, of course, only true in the context of a network in which transport units move at a nominally fixed speed and thus mean travel times for all origin/destination pairs are fixed.}\]
and accelerate unnecessarily in light traffic conditions — an undesirable feature for traffic control.

The second choice involved a quadratic functional of the same quantities. The resulting \( u_r \) is continuous but \( u_f \) is still of the bang-bang type except in a singular control interval. This is because only external smoothness, i.e., the smoothness between input and output, was considered. For smooth flow to result, it is necessary to also achieve internal smoothness; i.e., to minimize the difference between \( u_r \) and \( u_f \) as well as between \( u_f \) and \( r_D \). In turn, this should essentially minimize the difference between \( u_r \) and \( r_D \). Thus, the following quadratic objective function, which was developed in this study, was judged to be the most appropriate:

\[
J = \int_{t_o}^{t_f} \left[ q_1(t) + a_2 q_2(t) + a_3 S_f^2 (u_f(t) - r_D(t))^2 + a_4 S_D^2 (u_r(t) - u_f(t))^2 \right] dt. \tag{2.3}
\]

Here \( t_o \) and \( t_f \) are the times when the control law starts and stops, respectively, \( a_2, a_3 \) and \( a_4 \) are weighting factors, and \( S_f \) and \( S_D \), as will be determined later, are collective costs associated with the upstream and downstream links, respectively. Here, without loss of generality, \( t_o = 0 \) will be assumed. Further, \( t_f \) will be selected as the time when control law updating is scheduled (i.e., a new \( r_D \) is to be employed), and its value would depend on traffic conditions. A large \( t_f \), corresponding to infrequent updating, would be employed for light traffic and small changes in \( r_D \), and a small \( t_f \) for heavy traffic and/or large changes in \( r_D \). The latter choice should ameliorate the effect of estimation errors by insuring that congestion doesn't result. For infrequent updating, a free \( t_f \) would be employed whereas for frequent updating, a fixed \( t_f \) would be used.

Different values of \( a_2, a_3 \) and \( a_4 \) would be chosen for different traffic conditions and node geometries. For example, in heavy traffic \( a_2 \simeq 1 \) and \( a_3, a_4 \ll 1 \) would
be selected so that congestion could be cleared quickly. If $N_0$ lacked adequate storage, a small $a_2$ might be preferable and queue formation would be transferred to upstream nodes.

The problem is to find a solution that would minimize $J$. In general, optimal solutions may or may not exist, depending on the traffic demand.

2.4.1 Optimal Solutions

Initially, the case where $r_D < C_f$ and $r_D < C_r$ will be considered. In this case, the queues can be cleared and queue rearrangement is not necessary since both upstream and downstream links are capable of carrying the requested traffic. Hence, the origin is reachable and optimal controls exist. The Hamiltonian $H$ is defined as

$$H = q_1^2 + a_2 q_2^2 + a_3 S_f^2 (u_f - r_D)^2 + a_4 S_D^2 (u_r - u_f)^2 + p_1 (u_f - u_r) + p_2 (r_D - u_f),$$  \hspace{1cm} (2.4)$$

and the necessary conditions for an extremal become

$$\dot{p}_1 = -2q_1 \hspace{1cm} (2.5a)$$
$$\dot{p}_2 = -2a_2 q_2 \hspace{1cm} (2.5b)$$
$$H(q_1^*, q_2^*, u_f^*, u_r^*) = \min_{u_f, u_r} H(q_1, q_2, u_f, u_r) \hspace{1cm} (2.5c)$$

together with the dynamic equations (2.1a) – (2.1b). Here $p_1$ and $p_2$ are costates associated with $q_1$ and $q_2$, respectively, and $(\cdot)^*$ is the optimal function associated with $(\cdot)$.

In general, $r_D(t)$ would be an arbitrary time function; however, for analytical reasons, a constant $r_D$ will be considered here. Such a constant could be an average value, or an estimate of a stochastic process over the control interval. The latter is
determined locally at \( N_0 \) using only this quantity and \( q_1(0) \) and \( q_2(0) \). Two cases are possible: A) \( t_f \) free, and B) \( t_f \) fixed. In both cases, the queues may fall into the regions in which the following combination of controls would be necessary: 1) \( C_f > u_f \geq 0 \) and \( C_r > u_r \geq 0 \); 2) \( C_f > u_f \geq 0 \) and \( u_r = C_r \); 3) \( u_f = C_f \) and \( C_r > u_r \geq 0 \); and 4) \( u_f = C_f \) and \( u_r = C_r \).

A) \( t_f \) free

Since \( r_D \) is independent of \( t \), \( H \) would be independent of \( t \) explicitly, and equal to zero over \([0, t_f]\) [60,61].

1) \( C_f > u_f \geq 0 \) and \( C_r > u_r \geq 0 \)

Such a scenario arises when the traffic demand is less than the upstream and downstream capacities, the queue lengths are moderate, and the queues can be cleared without excessive input and output flows. Since saturation doesn’t exist, equation (2.5c) becomes

\[
\frac{\partial H}{\partial u_r} = 0 \tag{2.6a}
\]

\[
\frac{\partial H}{\partial u_f} = 0. \tag{2.6b}
\]

so that

\[
u_r = \frac{p_1}{2a_4S_D^2} + u_f \tag{2.7a}\]

and

\[
u_f = \frac{-p_1 + p_2 + 2a_3S_f^2r_D + 2a_4S_D^2u_r}{2a_3S_f^2 + 2a_4S_D^2} \tag{2.7b}\]

or,

\[
u_f = \frac{p_2}{2a_3S_f^2} + r_D. \tag{2.7c}\]

Substituting these equations into \( H \) yields

\[
H = q_1^2(t) + a_2q_2^2(t) - \frac{p_2^2}{4a_3S_f^2} - \frac{p_1^2}{4a_4S_D^2} \tag{2.8}\]
Since $H=0$,

\[ p_1 = 2\sqrt{a_4 S_D^2 q_1} \] (2.9a)

and

\[ p_2 = 2\sqrt{a_2 a_3 S_f^2 q_2}. \] (2.9b)

Substituting these equations into (2.7a) and (2.7c), one obtains

\[ u_f = \sqrt{\frac{a_2}{a_3 S_f^2}} q_2 + r_D \] (2.10a)

and

\[ u_r = \frac{q_1}{\sqrt{a_4 S_D^2}} + u_f = \frac{q_1}{\sqrt{a_4 S_D^2}} + \sqrt{\frac{a_2}{a_3 S_f^2}} q_2 + r_D, \] (2.10b)

so that

\[ \dot{q}_1 = \frac{-q_1}{\sqrt{a_4 S_D^2}} \] (2.11a)

and

\[ \dot{q}_2 = -\sqrt{\frac{a_2}{a_3 S_f^2}} q_2. \] (2.11b)

Defining $\alpha_1 \overset{\text{def}}{=} \sqrt{\frac{1}{a_4 S_D^2}}$ and $\alpha_2 \overset{\text{def}}{=} \sqrt{\frac{a_2}{a_3 S_f^2}}$ it follows that

\[ q_1(t) = q_1(0)e^{-\alpha_1 t} \] (2.12a)

\[ q_2(t) = q_2(0)e^{-\alpha_2 t} \] (2.12b)

\[ u_f(t) = \alpha_2 q_2(t) + r_D \] (2.12c)

and

\[ u_r(t) = \alpha_1 q_1(t) + \alpha_2 q_2(t) + r_D \] (2.12d)

That is, the controls that will clear the queues while minimizing $J$ are linear combinations of $r_D$ and the equivalent flows due to $q_1$ and $q_2$. Strictly speaking, optimal controls do not exist since the specified control cannot drive $q_1$ and $q_2$. 

26
to zero optimally in finite time. Nevertheless, since the controls are simple and queues can be considered as practically zero if they are less than a small constant, these controls are useful approximations. That is, \( t_f \) can be approximated by a finite number obtained from

\[
q_2^2(t_f) + a_2q_1^2(t_f) = \delta^2,
\]

where \( \delta \) is a small positive number. Substituting (2.12a) and (2.12b) into (2.13),

\[
q_1^2(0)e^{-2a_1t_f} + a_2q_2^2(0)e^{-2a_2t_f} = \delta^2
\]

results.

Defining

\[
\Omega_0 = \{q_1, q_2 : q_1 \geq 0, q_2 \geq 0\},
\]

\[
\Omega_1 = \{q_1, q_2 : a_2q_2 + r_D \leq C_f\},
\]

and

\[
\Omega_2 = \{q_1, q_2 : a_1q_1 + a_2q_2 + r_D \leq C_r\},
\]

the domain for the unsaturated controls is \( \Omega_0 \cap \Omega_1 \cap \Omega_2 \) as shown by the shaded area of Fig. 7. Note that \( q_1(t) \) and \( q_2(t) \) were needed in the specification of \( u_r \) and \( u_f \). These queues could be obtained by computations involving measured queue sizes and \( r_D(t) \).

2) \( C_f > u_f \geq 0 \) and \( u_r = C_r \)

Such a situation could arise after a rush-hour peak had passed and \( q_1 \) were still large. From 1), if \( q_1 \) and \( q_2 \) were not in \( \Omega_0 \cap \Omega_1 \cap \Omega_2 \), a saturated \( u_r \) would be necessary. The minimization of \( H \) is achieved by using equation (2.7b) and

\[
u_f = \frac{-p_1 + p_2 + 2a_3S_f^2r_D + 2a_4S_D^2C_r}{2a_3S_f^2 + 2a_4S_D^2}
\]

(2.15)
results. Defining \( \alpha \overset{\text{def}}{=} \sqrt{\frac{1}{a_3 S_f^2 + a_4 S_D^2}} \) and \( p \overset{\text{def}}{=} p_1 - p_2 \), it follows that

\[
uf = -\frac{\alpha^2}{2} p + \alpha^2 (a_3 S_f^2 r_D + a_4 S_D^2 C_r)
\]

(2.16)

and

\[
H = q_1^2 + a_2 q_2^2 - \frac{\alpha^2}{4} (p_1 - p_2)^2 + \\
\alpha^2 \left[ (p_1 a_3 S_f^2 + a_2 a_4 S_D^2) (r_D - C_r) + a_3 a_4 S_f^2 S_D^2 (r_D - C_r)^2 \right].
\]

(2.17)

Since \( H=0 \), \( p_1 \) and \( p_2 \) can be obtained as

\[
p_1 = \frac{-a_2 (q_1 + q_2)^2}{(1 + a_2) (r_D - C_r)} + \frac{2 (q_1 - a_2 q_2)}{\alpha a (1 + a_2)} + \\
\frac{(r_D - C_r) \left[ a_3 S_f^2 - a_2 (2 + a_2) a_4 S_D^2 \right]}{(1 + a_2)^2}
\]

(2.18)

and

\[
p_2 = \frac{-a_2 (q_1 + q_2)^2}{(1 + a_2) (r_D - C_r)} - \frac{2 (q_1 - a_2 q_2)}{\alpha a (1 + a_2)} + 
\]
Figure 8: Domain associated with Saturated $u_r$, $(1 + a_2) C_f \geq a_2 r_D + C_r$.

$$\frac{r_D - C_r}{(1 + a_2)^2} \left[ -(2a_2 + 1) \left( a_3 S_f^2 \right)^2 - (1 + 2a_2 - a_2^2) a_3 a_4 S_f^2 S_D^2 + (a_2 a_4 S_D^2)^2 \right], \quad (2.19)$$

$$u_f = -\frac{\alpha_a}{1 + a_2} (q_1 - a_2 q_2) + \frac{1}{1 + a_2} (a_2 r_D + C_r). \quad (2.21)$$

Here, the first bracketed term portrays a compromise between $q_1$ and $q_2$, and the second a compromise between $r_D$ and $C_r$. That is, if both $r_D$ and $C_r$ are fixed, $u_f$ would be a compromise between $q_1$ and $q_2$, and $r_D$ and $C_r$.

The corresponding domain would be $\Omega_0 \cap \Omega_2 \cap \Omega_3$ as shown in the shaded area of Fig. 8, where $\Omega_2$ is the complement of $\Omega_2$, and $\Omega_3$ is defined as

$$\Omega_3 = \left\{ q_1, q_2 : -q_1 + a_2 q_2 \leq \frac{(1 + a_2) C_f - (a_2 r_D + C_r)}{\alpha_a} \right\}. \quad (2.22)$$
Substituting $u_f$ into (2.1a) and (2.1b) and solving,

$$q_1(t) = \frac{1}{1 + a_2} \left[ (q_1(0) - a_2q_2(0)) e^{-\alpha_a t} + a_2 (r_D - C_r) t + a_2 (q_1(0) + q_2(0)) \right],$$

(2.23a)

and

$$q_2(t) = \frac{1}{1 + a_2} \left[ -(q_1(0) - a_2q_2(0)) e^{-\alpha_a t} + (r_D - C_r) t + (q_1(0) + q_2(0)) \right],$$

(2.23b)

result. Here, the state trajectory might first approach the boundary of either $\Omega_2$ or $\Omega_3$. In the former case, since $-q_1(t) + a_2q_2(t) = (-q_1(0) + a_2q_2(0)) e^{-\alpha_a t}$, the state trajectory would approach $q_1(t) = a_2q_2(t)$ asymptotically; i.e., it would never cross the boundary of $\Omega_3$ if $(1 + a_2) C_f \geq a_2r_D + C_r$ since $q_1(t) = a_2q_2(t)$ would be inside of $\Omega_3$. The duration $t_{1a}$ the trajectory remained in $\Omega_0 \cap \overline{\Omega_2} \cap \Omega_3$ could be found by substituting (2.23a) and (2.23b) into $\alpha_1 q_1(t_{1a}) + \alpha_2 q_2(t_{1a}) + r_D = C_r$ and solving for these quantities. Thus,

$$(\alpha_1 - \alpha_2) (q_1(0) - a_2q_2(0)) e^{-\alpha_a t_{1a}} + (a_2 \alpha_1 + \alpha_2) (r_D - C_r) t_{1a} =$$

$$(1 + a_2) (C_r - r_D) - (a_2 \alpha_1 + \alpha_2) (q_1(0) + q_2(0)).$$

(2.24)

After $t_{1a}$, unsaturated controls would be applied and the associated duration could be estimated by employing equation (2.14), with $q_1(0)$, $q_2(0)$, and $t_f$ replaced by $q_1(t_{1a})$, $q_2(t_{1a})$, and $t_f - t_{1a}$, respectively. Consequently, the control interval would be the sum of $t_{1a}$ and $t_f - t_{1a}$.

If $(1 + a_2) C_f < a_2r_D + C_r$, $q_1(t) = a_2q_2(t)$ would be "above" $\Omega_3$, as shown in Fig. 9, and the state trajectory might cross the boundary of either $\Omega_2$ or $\Omega_3$. If it would cross the boundary of $\Omega_2$, the discussion above would apply; if it would cross that of $\Omega_3$ at $t = t_{1b}$, saturated controls would then be applied until the trajectory would arrive at the boundary of either $\Omega_2$ or $\Omega_4$, which is to be defined in 3). In
the former, unsaturated controls would then be applied to clear the queues. In the latter, after the trajectory entered Ω₄, it would then arrive at the boundary of Ω₁ and unsaturated controls would then be employed to clear the queues. Either way, substituting (2.23a) and (2.23b) into $-q_1(t_{1b}) + a_2q_2(t_{1b}) = \frac{(1 + a_2)C_f - (a_2r_D + C_r)}{\alpha a}$, the boundary of Ω₃, and solving for $t_{1b}$, there results

$$t_{1b} = \frac{1}{\alpha a} \ln \left( \frac{\alpha a (q_1(0) - a_2q_2(0))}{(a_2r_D + C_r) - (1 + a_2)C_f} \right).$$

(2.25)

In this case, $t_f$ would be the sum of $t_{1b}$, the duration in the region associated with both saturated controls, and the time associated with both unsaturated controls.

3) $u_f = C_f$ and $C_r > u_r \geq 0$

This condition could also be associated with the trailing period of a rush hour and might happen only when $C_f < C_r$. Here, however, $q_2$ would be very large initially because of limited $C_f$. The task is to drive the queues until $t = t_2$ when $u_f$ comes out of saturation; i.e., when $q_1$ and $q_2$ arrive at the boundary of Ω₁. In this case,
H = q_1^2 + a_2 q_2^2 + \left[ \frac{a_2}{\alpha_2^2} (C_f - r_D) - p_2 \right] (C_f - r_D) - \frac{\alpha_1^2 p_1^2}{4}. \quad (2.26)

Employing \( H = 0 \) and equation (2.6a) results in

\[ u_r = \frac{\alpha_1^2}{2} p_1 + C_f. \quad (2.27) \]

Further,

\[ p_1 = \frac{2}{\alpha_1} q_1, \quad (2.28a) \]
\[ p_2 = \frac{a_2}{\alpha_2^2} (C_f - r_D) + \frac{a_2 q_2^2}{C_f - r_D}. \quad (2.28b) \]

Thus,

\[ u_r = \alpha_1 q_1 + C_f, \quad (2.28c) \]

and the corresponding domain is \( \Omega_0 \cap \Omega_1 \cap \Omega_4 \) as shown in the shaded area of Fig. 10, where \( \Omega_1 \) is the complement of \( \Omega_1 \), and \( \Omega_4 \) is defined by

\[ \Omega_4 = \{ q_1, q_2 : \alpha_1 q_1 + C_f \leq C_r \}. \quad (2.29) \]

Substituting (2.28c) and \( u_f = C_f \) into (2.1a) and (2.1b), and solving for \( q_1 \) and \( q_2 \) result in

\[ q_1(t) = q_1(0) e^{-\alpha_1 t} \quad (2.30a) \]

and

\[ q_2(t) = (r_D - C_f) t + q_2(0) \quad (2.30b) \]

Noting that \( q_2(t_2) \) must lie on the boundary of \( \Omega_1 \) so that \( q_2(t_2) = \frac{C_f - r_D}{\alpha_2} \) and employing (2.1a), one obtains

\[ t_2 = \frac{q_2(0)}{C_f - r_D} - \frac{1}{\alpha_2}. \quad (2.31) \]
Here, $t_2$ is completely specified by $r_D, C_f, \text{ and } q_2(0)$, and is not directly affected by $u_r$. Therefore, the control interval can be determined by summing $t_2$ and $t_f - t_2$, with the latter being the duration in $\Omega_0 \cap \Omega_1 \cap \Omega_2$.

4) $u_f = C_f$ and $u_r = C_r$

This could occur after the peak of a rush hour. The corresponding domain would be $\Omega_0 \cap \Omega_2 \cap \Omega_3 \cap \Omega_4$, as shown in Fig. 11. The dynamics of the queues would be

\[
q_1(t) = (C_f - C_r) t + q_1(0) \quad (2.32a)
\]

and

\[
q_2(t) = (r_D - C_f) t + q_2(0). \quad (2.32b)
\]

Further, the time when at least one of the controls became unsaturated would be completely determined by $r_D, C_f, C_r, q_1(0), \text{ and } q_2(0)$.

There are three possibilities: $u_f$ and $u_r$ would become unsaturated simultaneously, $u_f$ would become unsaturated first, and $u_r$ would become unsaturated first,
as shown in Fig. 12. In the first, the goal would be the boundary of $\Omega_2$. Upon substituting (2.32a) and (2.32b) into $\alpha_1 q_1(t_{3a}) + \alpha_2 q_2(t_{3a}) + r_D = C_r$, one would obtain

$$t_{3a} = \frac{(r_D - C_r) + \alpha_1 q_1(0) + \alpha_2 q_2(0)}{\alpha_1 (C_r - C_f) + \alpha_2 (C_f - r_D)},$$

(2.33)

where $t_{3a}$ is the time when the trajectory would arrive at this boundary. The corresponding control interval would be the sum of $t_{3a}$ and $t_f - t_{3a}$ with the latter being the duration when both controls would be unsaturated.

If $u_f$ became unsaturated first, the goal would be the boundary of $\Omega_3$. Since the slope of the trajectory would be $\frac{C_f - r_D}{C_r - C_f}$ whereas that of the boundary of $\Omega_3$ is $\frac{1}{\alpha_2}$, the trajectory would approach $\Omega_3$ only when $(1 + a_2)C_f > a_2 r_D + C_r$. The time $t_{3b}$ for the state trajectory to go from an arbitrary state in this region to the boundary of $\Omega_3$ could be solved by substituting (2.32a) and (2.32b) into

$$-q_1(t_{3b}) + a_2 q_2(t_{3b}) = \frac{1}{\alpha_a} [(1 + a_2)C_f - a_2 r_D - C_r],$$

(2.34)

to yield
Figure 12: Possible trajectories in the Region Associated with Saturated Controls.

\[ t_{3b} = \frac{q_1(0) - a_2q_2(0)}{a_2r_D + C_r - (1 + a_2)C_f} - \frac{1}{\alpha_a}, \]  

(2.35)

After the trajectory entered \( \Omega_0 \cap \overline{\Omega_2} \cap \Omega_3 \), it would remain in that region until \( u_r \) would also become unsaturated. Thus, the control interval would be the sum of \( t_{3b} \), the duration in \( \Omega_0 \cap \Omega_1 \cap \overline{\Omega_2} \cap \Omega_3 \), and that in \( \Omega_0 \cap \Omega_1 \cap \Omega_2 \).

If \( u_r \) became unsaturated first, \( t_{3c} \) could be obtained by substituting (2.32a) into

\[ \alpha_1 q_1(t_{3c}) + C_f = C_r \]  

(2.36)

and

\[ t_{3c} = \frac{q_1(0)}{C_r - C_f} - \frac{1}{\alpha_1} \]  

(2.37)

would result. Similarly, the control interval would be the sum of \( t_{3c} \), the duration in \( \Omega_0 \cap \overline{\Omega_1} \cap \Omega_4 \), and that in \( \Omega_0 \cap \Omega_1 \cap \Omega_2 \).

The determination of which case would occur first could be accomplished by computing \( t_{3a} \), \( t_{3b} \), and \( t_{3c} \) apriori and selecting the case associated with the
smallest positive value. Here, if \((1 + a_2) C_f > a_2 r_D + C_r\), all three cases would be possible; if \((1 + a_2) C_f \geq a_2 r_D + C_r\), only the first and the last cases would be possible since \(t_{3b}\) would be either negative or infinity; and if \(C_r < C_f\), only the first two cases would be possible since \(t_{3c}\) would be negative.

If \((1 + a_2) C_f > a_2 r_D + C_r\), \(t_{3a} < t_{3b}\) and \(t_{3a} < t_{3c}\), the trajectory would move to the boundary of \(\Omega_2\) and the corresponding \(q_1(0)\) and \(q_2(0)\) would be in the region \(\Omega_0 \cap \Omega_2 \cap \Omega_3 \cap \Omega_4 \cap \Omega_5 \cap \Omega_6\), with \(\Omega_5\) and \(\Omega_6\) defined by

\[
\Omega_5 = \left\{ q_1(0), q_2(0) : \frac{(a_2 \alpha_1 + \alpha_2) \left[ (C_f - r_D) q_1(0) + (C_f - C_r) q_2(0) \right]}{\alpha_1 (C_r - C_f) + \alpha_2 (C_f - r_D)} \geq \frac{1}{\alpha_a} + \frac{r_D - C_r}{\alpha_1 (C_r - C_f) + \alpha_2 (C_f - r_D)} \right\}
\]

and

\[
\Omega_6 = \left\{ q_1(0), q_2(0) : \frac{\alpha_2 q_2(0)}{\alpha_1 (C_r - C_f) + \alpha_2 (C_f - r_D)} < \frac{C_r - r_D}{\alpha_1 (C_r - C_f) + \alpha_2 (C_f - r_D)} - \frac{1}{\alpha_1} \right\}.
\]

If \((1 + a_2) C_f > a_2 r_D + C_r\), \(t_{3a} > t_{3b}\) and \(t_{3b} < t_{3c}\), the trajectory would move to the boundary of \(\Omega_3\) and the corresponding \(q_1(0)\) and \(q_2(0)\) would be in the region specified by \(\Omega_0 \cap \Omega_2 \cap \Omega_3 \cap \Omega_4 \cap \Omega_5 \cap \Omega_7\), where \(\Omega_7\) is defined by

\[
\Omega_7 = \left\{ q_1(0), q_2(0) : \frac{a_2 \left[ (r_D - C_f) q_1(0) + (C_r - C_f) q_2(0) \right]}{(a_2 r_D + C_r) - (1 + a_2) C_f} \geq \frac{1}{\alpha_1} - \frac{1}{\alpha_a} \right\}.
\]

If \((1 + a_2) C_f > a_2 r_D + C_r\), \(t_{3a} > t_{3b}\) and \(t_{3b} > t_{3c}\), the trajectory would move to the boundary of \(\Omega_4\) and the corresponding \(q_1(0)\) and \(q_2(0)\) would be in the region specified by \(\Omega_0 \cap \Omega_2 \cap \Omega_3 \cap \Omega_4 \cap \Omega_5 \cap \Omega_6 \cap \Omega_7\).

If \((1 + a_2) C_f < a_2 r_D + C_r\), only \(t_{3a}\) and \(t_{3c}\) need to be compared. If \(t_{3a} < t_{3c}\), the trajectory would move to the boundary of \(\Omega_2\) and the corresponding \(q_1(0)\) and \(q_2(0)\) would be in the region \(\Omega_0 \cap \Omega_2 \cap \Omega_3 \cap \Omega_4 \cap \Omega_6\); if \(t_{3a} > t_{3c}\), the trajectory would
move to the boundary of $\Omega_4$ and the corresponding $q_1(0)$ and $q_2(0)$ would be in the region specified by $\Omega_0 \cap \overline{\Omega_2} \cap \overline{\Omega_3} \cap \overline{\Omega_4} \cap \overline{\Omega_6}$.

If $C_r < C_f$, only $t_{3a}$ and $t_{3b}$ need to be compared. If $t_{3a} < t_{3b}$, the trajectory would move to the boundary of $\Omega_2$ and the corresponding $q_1(0)$ and $q_2(0)$ would be in the region $\Omega_0 \cap \overline{\Omega_2} \cap \overline{\Omega_3} \cap \overline{\Omega_4} \cap \Omega_5$; if $t_{3a} > t_{3b}$, the trajectory would move to the boundary of $\Omega_3$ and the corresponding $q_1(0)$ and $q_2(0)$ would be in the region $\Omega_0 \cap \overline{\Omega_2} \cap \overline{\Omega_3} \cap \overline{\Omega_4} \cap \overline{\Omega_5}$.

In summary, the $q_1 - q_2$ space would be divided into several regions, with each corresponding to a combination of $u_f$ and $u_r$. Three types of partitioning are possible, as shown in Figs. 13 — Fig 15, corresponding to the relationships between $C_r$ and $C_f$ and $(1 + a_2)C_f$ and $C_r + a_2r$. In Fig. 13, where $C_r \geq C_f$ and $(1 + a_2)C_f \geq (C_r + a_2r)$, the possible control sequences are: { both unsaturated }, { unsaturated $u_r$ and saturated $u_f$, both unsaturated }, { saturated $u_r$ and unsaturated $u_f$, both unsaturated }, { both saturated, both unsaturated }, { both saturated, saturated $u_r$ and unsaturated $u_f$, both unsaturated }, and { both saturated, unsaturated $u_r$ and saturated $u_f$, both unsaturated }.

In Fig. 14, where $C_r \geq C_f$ and $(1 + a_2)C_f < (C_r + a_2r)$, the possible control sequences are: { both unsaturated }, { saturated $u_r$ and unsaturated $u_f$, both unsaturated }, { saturated $u_r$ and unsaturated $u_f$, both saturated, both unsaturated }, { saturated $u_r$ and unsaturated $u_f$, both saturated, unsaturated $u_r$ and saturated $u_f$, both unsaturated }, { unsaturated $u_r$ and saturated $u_f$, both unsaturated }, { both saturated, both unsaturated }, and { both saturated, unsaturated $u_r$ and saturated $u_f$, both unsaturated }.

In Fig. 15, which corresponds to the case $C_r < C_f$, only three domains are possible. This results from a negative $\frac{C_r - C_f}{a_1}$, which is the boundary of $\Omega_4$. Physically, this means that when the downstream capacity is limited, the upstream
capacity can not be saturated without saturating the former. The possible control sequences are: { both unsaturated }, { saturated ur and unsaturated uf, both unsaturated }, { both saturated, both unsaturated }, and { both saturated, saturated ur and unsaturated uf, both unsaturated }.

B) $t_f$ fixed

Although the controls associated with free $t_f$ would clear queues, it might be desirable to clear queues in a specified time; e.g., before a rush hour or when the traffic demand is increasing. Again, four cases are possible for the first combination of controls: both unsaturated, $u_r$ saturated and $u_f$ unsaturated, $u_r$ unsaturated and $u_f$ saturated, and both saturated. Here, the reachable set $\Omega_r$, i.e., the set of the initial queue sizes that can be cleared in $t_f$ time units, can be determined by letting $u_r = C_r$ and $u_f = C_f$ when both queues are not zero, $u_r = C_r$ and $u_f = r_D$ when $q_2(t) = 0$ and $q_1(t)$ is not zero, and $u_r = u_f = C_f$ when $q_1(t) = 0$. 

Figure 13: Domains for Controls — case a.
and $(1+a_2)c_f < (c_c+a_2\gamma_D) c_f$

Both sat.

Both unsat.

Both sat.

Both unsat.

Figure 14: Domains for Controls — case b.

Figure 15: Domains for Controls — case c.
and $q_2(t)$ is not zero, and solving for $q_1(0)$ and $q_2(0)$. Thus,

$$\Omega_r = \{q_1(0), q_2(0) : q_1(0) + q_2(0) \leq (C_r - r_D) t_f \text{ and } q_2(0) \leq (C_f - r_D) t_f\}.$$ 

In the sequel, the initial queue sizes will be assumed to satisfy this condition.

1) $C_f > u_f \geq 0$ and $C_r > u_r \geq 0$

Here, the optimal controls are obtained by using equations (2.5a), (2.5b), (2.7a) and (2.7c). These result in

$$u_f = r_D - \alpha_2^2 \int_0^t q_2(\tau) d\tau + \frac{\alpha_2^2}{2a_2} p_2(0)$$

and

$$u_r = u_f - \alpha_1^2 \int_0^t q_1(\tau) d\tau + \frac{\alpha_1^2}{2} p_1(0),$$

or

$$u_r = r_D - \alpha_2^2 \int_0^t q_2(\tau) d\tau - \alpha_1^2 \int_0^t q_1(\tau) d\tau + \frac{\alpha_2^2}{2a_2} p_2(0) + \frac{\alpha_1^2}{2} p_1(0).$$

Taking derivatives on both sides of equations (2.5a) and (2.5b) and employing equations (2.7a) and (2.7c), together with the state equations, one obtains

$$\ddot{p}_1 = \alpha_1^2 p_1$$

and

$$\ddot{p}_2 = \alpha_2^2 p_2.$$ 

Thus,

$$p_1(t) = b_1 e^{\alpha_1 t} + b_2 e^{-\alpha_1 t}$$

and

$$p_2(t) = b_3 e^{\alpha_2 t} + b_4 e^{-\alpha_2 t},$$

so that

$$q_1(t) = \frac{-\alpha_1}{2} \left[ b_1 e^{\alpha_1 t} - b_2 e^{-\alpha_1 t} \right].$$
\[
q_2(t) = \frac{-\alpha_2}{2\alpha_2} \left[ b_3 e^{\alpha_2 t} - b_4 e^{-\alpha_2 t} \right].
\] (2.41b)

The quantities \(b_1, b_2, b_3\) and \(b_4\) are easily expressed in terms of \(q_1(0), q_2(0)\) and \(q_1(t_f) = 0\) and \(q_2(t_f) = 0\). In general, \(q_1\) and \(q_2\) do not have to become zero simultaneously; however, since both queues could be emptied at approximately the same time and this would result in a simplification of the mathematics, this condition will be employed. Thus,

\[
b_1 = \frac{2q_1(0)}{\alpha_1 \left[ e^{2\alpha_1 t_f} - 1 \right]},
\] (2.42a)

\[
b_2 = \frac{2q_1(0)}{\alpha_1 \left[ 1 - e^{-2\alpha_1 t_f} \right]},
\] (2.42b)

\[
b_3 = \frac{2a_2 q_2(0)}{\alpha_2 \left[ e^{2\alpha_2 t_f} - 1 \right]},
\] (2.42c)

and

\[
b_4 = \frac{2a_2 q_2(0)}{\alpha_2 \left[ 1 - e^{-2\alpha_2 t_f} \right]}.
\] (2.42d)

Then

\[
p_1(0) = b_1 + b_2 = \frac{2q_1(0) \left[ 1 + e^{2\alpha_1 t_f} \right]}{\alpha_1 \left[ e^{2\alpha_1 t_f} - 1 \right]} = \frac{2q_1(0)}{\alpha_1} \coth (\alpha_1 t_f),
\] (2.43a)

and

\[
p_2(0) = b_3 + b_4 = \frac{2a_2 q_2(0) \left[ 1 + e^{2\alpha_2 t_f} \right]}{\alpha_2 \left[ e^{2\alpha_2 t_f} - 1 \right]} = \frac{2a_2 q_2(0)}{\alpha_2} \coth (\alpha_2 t_f).
\] (2.43b)

Therefore,

\[
u_f = r_D - \alpha_2^2 \int_0^t q_2(\tau)d\tau + a_2 q_2(0) \coth (\alpha_2 t_f) \quad 0 \leq t \leq t_f, \quad (2.44a)
\]
and

\[ u_r = r_D - \alpha_1^2 \int_0^t q_1(\tau) d\tau - \alpha_2^2 \int_0^t q_2(\tau) d\tau + \alpha_1 q_1(0) \coth(\alpha_1 t_f) + \alpha_2 q_2(0) \coth(\alpha_2 t_f) \quad 0 \leq t \leq t_f. \tag{2.44b} \]

The boundary within which these controls are optimal can be determined by letting \( u_f = C_f \) and \( u_r = C_r \) in equations (2.38a) and (2.38b) at \( t = 0 \) when the controls are maximal, and rearranging. Thus,

\[ r_D + \alpha_2 \coth(\alpha_2 t_f) q_2(0) \leq C_f \tag{2.45a} \]

and

\[ r_D + \alpha_1 \coth(\alpha_1 t_f) q_1(0) + \alpha_2 \coth(\alpha_2 t_f) q_2(0) \leq C_r. \tag{2.45b} \]

Note that the integral terms are not included since they are zero at \( t = 0 \).

Defining

\[ \Omega'_1 = \{ q_1, q_2 : r_D + \alpha_2 \coth(\alpha_2 t_f) q_2(0) \leq C_f \} \]

and

\[ \Omega'_2 = \{ q_1, q_2 : r_D + \alpha_1 \coth(\alpha_1 t_f) q_1(0) + \alpha_2 \coth(\alpha_2 t_f) q_2(0) \leq C_r \}, \]

the domain in which the optimal controls (2.38a) and (2.38b) are to be applied is \( \Omega_0 \cap \Omega'_1 \cap \Omega'_2 \), as shown in Fig. 16. Note that \( \Omega'_i \) becomes \( \Omega_i, i = 1, 2 \) when \( t_f \) approaches \( \infty \) as would be expected.

2) \( C_f > u_f \geq 0 \) and \( u_r = C_r \)

From 1), if \( q_1 \) and \( q_2 \) were not in \( \Omega_0 \cap \Omega'_1 \cap \Omega'_2 \), a saturated control would be necessary. The costates and states could be obtained by taking time derivatives on both sides of (2.5a) and (2.5b) and employing the state equations (2.1a) and (2.1b). Thus,
Figure 16: Domain Associated with Unsaturated Controls, $t_f$ fixed.

\[ p_1 = \alpha^2 \left[ p_1 - p_2 - 2a_3S_f^2(r_D - C_r) \right] \quad (2.46) \]

and

\[ p_2 = -a_2\alpha^2 \left[ p_1 - p_2 + 2a_4S_D^2(r_D - C_r) \right]. \quad (2.47) \]

so that

\[ p_1 = b_5e^{\alpha a t} + b_6e^{-\alpha a t} - \frac{a_2(r_D - C_r)}{1 + a_2} t^2 + b_7 t + b_8 + \]

\[ \frac{2(r_D - C_r) \left( a_3S_f^2 - a_2a_4S_D^2 \right)}{(1 + a_2)^2}, \quad (2.48a) \]

\[ p_2 = -a_2b_5e^{\alpha a t} - a_2b_6e^{-\alpha a t} - \frac{a_2(r_D - C_r)}{1 + a_2} t^2 + b_7 t + b_8 + \]

\[ \frac{2a_2(r_D - C_r) \left( -a_3S_f^2 + a_2a_4S_D^2 \right)}{(1 + a_2)^2}, \quad (2.48b) \]

\[ p = (1 + a_2) \left( b_5e^{\alpha a t} + b_6e^{-\alpha a t} \right) + \frac{2(r_D - C_r) \left( a_3S_f^2 - a_2a_4S_D^2 \right)}{(1 + a_2)}, (2.48c) \]

\[ q_1 = -\frac{\alpha a}{2} \left( b_5e^{\alpha a t} - b_6e^{-\alpha a t} \right) + \frac{a_2(r_D - C_r)t}{1 + a_2} - \frac{b_7}{2}, \quad (2.48d) \]

and
where \( p \) was defined in page 28. From equations (2.5a), (2.5b), and (2.16), \( u_f \) can be determined as

\[
u_f = \alpha^2 \int_0^t (q_1(\tau) - a_2 q_2(\tau)) d\tau - \frac{a_2^2 (b_5 + b_6)}{2} + \frac{a_2 r_D + C_r}{1 + a_2}.
\]

(2.49)

The corresponding domain is \( \Omega_0 \cap \Omega_2' \cap \Omega_3' \), where \( \Omega_2' \) is the complement of \( \Omega_2 \) and \( \Omega_3' \) is defined by

\[
\Omega_3' = \left\{ q_1(0), q_2(0) : -\frac{\alpha^2}{2} p + \alpha^2 \left( a_3 S_f^2 r_D + a_4 S_D^2 C_r \right) \leq C_f \right\}.
\]

Since this is in terms of \( p \), it is not easy to use; thus, \( \Omega_3' \) will be obtained in terms of \( q_1 \) and \( q_2 \) as is now described. Assuming \( b_5 \) and \( b_6 \) are of the same sign and subtracting \( a_2 q_2 \) from \( q_1 \) in (2.48d) and (2.48e), one obtains

\[
\frac{-2}{\alpha_a (1 + a_2) K_1} (q_1 - a_2 q_2) = \sinh (a_a t + \varphi),
\]

(2.50)

where \( K_1 = b_5 b_6 \) and \( \varphi = \frac{1}{2} \ln \frac{b_5}{b_6} \). Further, from (2.48c) it follows that

\[
\frac{p - K_2}{(1 + a_2) K_1} = \cosh (a_a t + \varphi),
\]

(2.51)

where \( K_2 = \frac{2(r_D - C_r)(a_3 S_f^2 - a_2 a_4 S_D^2)}{1 + a_2} \). Squaring (2.50) and (2.51) and noting that \( \cosh^2 x = 1 + \sinh^2 x \) yields

\[
p = K_2 \pm \sqrt{\frac{4(q_1 - a_2 q_2)^2}{\alpha_a^2} + (1 + a_2)^2 K_1^2}.
\]

(2.52)

Substituting (2.52) into \( \Omega_3' \) and rearranging results in

\[
q_1 - a_2 q_2 \geq K_4,
\]

(2.53)
Figure 17: Domain associated with Saturated $u_r$, $t_f$ fixed.

where

$$K_4 = \begin{cases} 0 & \text{if } \frac{a_2 r_D + C_r - (1 + a_2) C_f}{\alpha_a^2} \leq (1 + a_2)^2 K_1^2 \\ \pm \frac{a_2}{2} \sqrt{a_2 r_D + C_r - (1 + a_2) C_f - (1 + a_2)^2 K_1^2} & \text{otherwise.} \end{cases}$$

If $b_5$ and $b_6$ are different in sign, the same procedure can be followed to obtain

$$q_1 - a_2 q_2 \geq K_5,$$

(2.54)

where

$$K_5 = \begin{cases} 0 & \text{if } \frac{a_2 r_D + C_r - (1 + a_2) C_f}{\alpha_a^2} \leq (1 + a_2)^2 K_1^2 \\ \pm \frac{a_2}{2} \sqrt{a_2 r_D + C_r - (1 + a_2) C_f + (1 + a_2)^2 K_1^2} & \text{otherwise.} \end{cases}$$

Thus, $\Omega_3'$ can be described by

$$\Omega_3' = \{q_1, q_2 : q_1 - a_2 q_2 \geq K\},$$

as shown in Fig. 17, where $K$ may be either $K_4$ or $K_5$, depending on the signs of $b_5$ and $b_6$. 

45
3) $u_f = C_f$ and $C_r > u_r \geq 0$

Since $u_r$ cannot affect $q_2$, $t_2$ can be obtained from (2.31) by replacing $\alpha_2$ by $\alpha_2 \coth (\alpha_2 (t_f - t_2))$, and a fixed $t_f$ can be achieved by adjusting $t_f - t_2$. The optimal value of this quantity can be obtained by considering $H_c(t_2) = H_a \left( t_f - t_2 \right)$. The control $u_r$ is the same as (2.28c), and the corresponding domain is $\Omega_0 \cap \Omega_1^r \cap \Omega_4$, where $\Omega_1^r$ is the complement of $\Omega_1^r$.

4) $u_f = C_f$ and $u_r = C_r$

When both controls are saturated, the state trajectory is determined by (2.32a) and (2.32b), and the domain for saturated controls is $\Omega_0 \cap \Omega_2^r \cap \Omega_3^r \cap \Omega_4$.

It is now convenient, for purposes of explanation, to employ a descriptive notation for the queues, the costates, and the Hamiltonian in each region. An additional subscript, $a, b, c$, or $d$, will be added to each term to denote the region to which it applies; e.g., $a, b, c$, and $d$ will correspond to the region associated with unsaturated controls, saturated $u_r$ and unsaturated $u_f$, unsaturated $u_r$ and saturated $u_f$, and saturated controls, respectively.

Similar to the free $t_f$ case, if the state point is initially in $\Omega_0 \cap \Omega_2^r \cap \Omega_3^r$, the applied controls would either drive it to $\Omega_2^r$ or to a region where both controls would be saturated, as shown in Fig. 18. In the former, the Weierstrass-Erdmann corner condition [70] dictates that

$$p_{1b}(t_{1a}) = p_{1a}(t_{1a}), \quad (2.55a)$$

$$p_{2b}(t_{1a}) = p_{2a}(t_{1a}), \quad (2.55b)$$

and

$$H_b(t_{1a}) = H_a(t_{1a}), \quad (2.55c)$$

where $t_{1a}$ is the time when the trajectory would arrive at $\Omega_2^r$. Further, since the trajectory must be continuous on the boundary of $\Omega_2^r$, the queues in both regions
must satisfy

\[ q_{1b}(t_{1a}) = q_{1a}(t_{1a}) \]  \hspace{1cm} (2.56a)

and

\[ q_{2b}(t_{1a}) = q_{2a}(t_{1a}). \]  \hspace{1cm} (2.56b)

That is, if initially, the states were in \( \Omega_0 \cap \overline{\Omega_2^2} \cap \Omega_3^3 \), and the boundary of \( \Omega_2^2 \) were reached first, the trajectory would be determined by (2.48a) – (2.48e), and the nine unknowns \( b_5, b_6, b_7, b_8, t_{1a}, b_1, b_2, b_3, \) and \( b_4 \) would be determined by the nine equations given by \( q_1(0) = q_{1b}(0), q_2(0) = q_{2b}(0) \), the Weierstrass-Erdmann corner conditions given by (2.55a) – (2.55c), the boundary conditions on \( \Omega_2^2 \) given by (2.56a) and (2.56b), and the terminal conditions \( q_{1a}(t_f) = 0 \) and \( q_{2a}(t_f) = 0 \).

If the trajectory arrived at the boundary of \( \Omega_3^3 \) at \( t = t_{1b} \), the subsequent trajectory would proceed to either the boundary of \( \Omega_2' \) (see trajectory 2) or to that of \( \Omega_4 \) (see trajectory 3). In the latter case, the trajectory would then move to the boundary of \( \Omega_1' \) and both unsaturated controls would be employed thereafter. In
the first case, the 14 unknowns \( b_5, b_6, b_7, b_8, t_{1b}, b_{13}, b_{14}, b_{15}, b_{16}, t_{s2}, b_1, b_2, b_3, \) and \( b_4 \) would be determined by \( q_1(0) = q_{16}(0), q_2(0) = q_{26}(0) \), the Weierstrass-Erdmann conditions

\[
\begin{align*}
 p_{1b}(t_{1b}) &= p_{1d}(t_{1b}), \\ p_{2b}(t_{1b}) &= p_{2d}(t_{1b}), \\ H_b(t_{1b}) &= H_d(t_{1b}), \\ p_{1d}(t_{s2}) &= p_{1a}(t_{s2}), \\ p_{2d}(t_{s2}) &= p_{2a}(t_{s2}), \\ H_d(t_{s2}) &= H_a(t_{s2}),
\end{align*}
\] (2.57a)

the boundary conditions

\[
\begin{align*}
 q_{1b}(t_{1b}) &= q_{1d}(t_{1b}), \\ q_{2b}(t_{1b}) &= q_{2d}(t_{1b}), \\ q_{1d}(t_{s2}) &= q_{1a}(t_{s2}), \\ q_{2d}(t_{s2}) &= q_{2a}(t_{s2}),
\end{align*}
\] (2.58a)

and the terminal conditions \( q_{1a}(t_f) = 0 \) and \( q_{2a}(t_f) = 0 \). Here, \( t_{s2} \) is the time the trajectory would arrive at the boundary of \( \Omega_2' \) with both saturated controls applied, and \( b_{13}, b_{14}, b_{15}, \) and \( b_{16} \) are the corresponding constants of integration.

If the trajectory reached the boundary of \( \Omega_3' \) at \( t_{1c} \) first, followed by that of \( \Omega_4' \), then that of \( \Omega_1' \), the 19 unknowns \( b_5, b_6, b_7, b_8, t_{1c}, b_{13}, b_{14}, b_{15}, b_{16}, t_{s3}, b_9, b_{10}, b_{11}, b_{12}, t_{s4}, b_1, b_2, b_3, \) and \( b_4 \) would be determined by \( q_1(0) = q_{16}(0), q_2(0) = q_{26}(0) \), the Weierstrass-Erdmann corner conditions

\[
\begin{align*}
 p_{1b}(t_{1c}) &= p_{1d}(t_{1c}), \\ p_{2b}(t_{1c}) &= p_{2d}(t_{1c}),
\end{align*}
\] (2.59a)
\[ H_b(t_{1c}) = H_d(t_{1c}), \]  
\[ p_{1d}(t_{s3}) = p_{1c}(t_{s3}), \]  
\[ p_{2d}(t_{s3}) = p_{2c}(t_{s3}), \]  
\[ H_d(t_{s3}) = H_c(t_{s3}), \]  
\[ p_{1c}(t_{s4}) = p_{1a}(t_{s4}), \]  
\[ p_{2c}(t_{s4}) = p_{2a}(t_{s4}), \]  
\[ H_c(t_{s4}) = H_a(t_{s4}), \]  

the boundary conditions

\[ q_{1b}(t_{1c}) = q_{1d}(t_{1c}), \]  
\[ q_{2b}(t_{1c}) = q_{2d}(t_{1c}), \]  
\[ q_{1d}(t_{s3}) = q_{1c}(t_{s3}), \]  
\[ q_{2d}(t_{s3}) = q_{2c}(t_{s3}), \]  
\[ q_{1c}(t_{s4}) = q_{1a}(t_{s4}), \]  
\[ q_{2c}(t_{s4}) = q_{2a}(t_{s4}), \]  

and the terminal conditions \( q_{1a}(t_f) = 0 \) and \( q_{2a}(t_f) = 0 \). Here, \( t_{s3} \) is the time when the trajectory would arrive at the region in which saturated controls would be employed, \( b_9, b_{10}, b_{11} \) and \( b_{12} \) are the corresponding constants of integration, and \( t_{s4} \) is the time when the trajectory would arrive at the region where both controls were unsaturated.

If the initial states are in \( \Omega_0 \cap \Omega_2' \cap \Omega_3' \cap \Omega_4' \), the durations for the queues to leave this region are given by (2.33) – (2.37), depending on which of the following boundaries the trajectory would reach first: \( \Omega_2', \Omega_3', \) or \( \Omega_4' \). In the first case, unsaturated controls would be applied subsequently until the queues were cleared.
In the second, the trajectory would then move to the boundary of $\Omega'_2$. In the last case, it would then move to the boundary of $\Omega'_1$ and unsaturated controls would be applied subsequently.

### 2.4.2 Optimal Controls Do Not Exist

If $r_D \geq C_f$ or $r_D \geq C_r$, the origin would be unreachable and optimal controls would not exist. A reasonable policy would be to reduce the queue build-up at $N_0$. This would involve two possibilities: i) $C_f \geq C_r$; and ii) $C_f < C_r$. In the first, $u_r = C_r$ would be selected. One could select $u_f = C_r$; however, either inappropriate utilization of node storage or serious congestion might result since $q_1$ might remain too small or too large. Alternatively, one might drive $q_1$ to a preset value $q_{1s}$ and then let $u_f = u_r = C_r$. Thus, the problem would be to minimize

$$J_{x1} = \int_{0}^{t_{x1}} \left\{ (q_1 - q_{1s})^2 + a_4 S_D^2 (C_r - u_f)^2 \right\} dt$$

(2.61)

subject to the $q_1$ dynamic constraint (2.1a) and the positivity constraints (2.2a) - (2.2e). Here, the first term corresponds to "setpoint" regulation whereas the second is associated with smoothness. The quantity $t_{x1}$ is the time when $r_D \leq C_r$ and optimal controls can be applied. Upon solving for $u_f$ and accounting for its limitation $C_f$, one obtains

$$u_f = \min \left\{ C_r - \alpha_1 (q_1 - q_{1s}), C_f \right\}.$$  

(2.62)

In the second case, $u_f = C_f$ would be a reasonable solution and $u_r$ would be determined by considering $q_1$ only since $u_r$ would not affect $q_2$ directly. Thus,

$$\dot{q}_1 = C_f - u_r$$

(2.63)

and the problem would be to minimize
\[ J_{x2} = \int_0^{t_{x2}} \left\{ q_1^2 + a_4 S_D^2 (u_r - C_f)^2 \right\} dt, \]  

(2.64)

where \( t_{x2} \) is the time when \( r_D \leq C_f \). Solving for \( u_r \) and incorporating its maximum value, one obtains

\[ u_r = \min \{ \alpha_1 q_1 + C_f, C_r \}. \]  

(2.65)

### 2.4.3 Allocation of Input and Output Traffic

The design of an allocation scheme for upstream and downstream traffic involves various factors such as the type of traffic, the traffic demand from each upstream node, the cost associated with each downstream link, any priority assignment associated with each link, consideration of any emergency requirements, and other application-dependent criteria. Here, since the traffic is assumed to consist of automatically controlled automobiles operating under normal (nonemergency) conditions, only traffic demand, downstream costs, and priority assignment will be considered. The latter pertains to each node where, in general, traffic can arrive from an external entry facility and on links from upstream nodes. Usually, a higher priority would be assigned to traffic from the latter.

The cost, \( S_{f,i} \), associated with upstream link \( i \), is defined as

\[ S_{f,i} \overset{\text{def}}{=} \frac{f_d}{r_D,i}, \]  

(2.66)

where \( f_d = C_f \) to account for the maximum possible value of \( r_D,i \).

The aggregated upstream cost is defined as:

\[ S_f \overset{\text{def}}{=} \frac{f_d}{\sum_{i=1}^k w_i r_D,i}, \]  

(2.67)

where \( w_i \) is a weighting factor representing the priority assignment associated with link \( i \). Here, the \( w_i \) will be selected to satisfy \( \sum_{i=1}^k w_i = k \); thus for equal
weighting, \( w_i = 1, i = 1, \ldots, k \). An entry ramp will be viewed as an upstream link which generally has a low priority. Further, if \( r_{D,i} = 0 \) for some \( i \), that link will not be considered. Thus, the updated \( S_f \) to be used in the computation of the next controls at \( N_0 \) would be

\[
\frac{1}{S_f} = \sum_{i=1}^{k} \frac{w_i}{S_{f,i}}.
\]

(2.68)

Two approaches for upstream traffic allocation have been considered: first serving the nodes with the least cost, and serving all nodes in proportion to their cost. In the first, all (or the maximum amount feasible) of the requested traffic at the node associated with the lowest cost would be allowed to proceed to \( N_0 \), then the maximum amount feasible from the next lowest cost node, etc., until \( u_f \) were completely assigned. However, vehicles waiting at a higher-cost node might encounter excessive delay. Further, traffic flow might not be smooth since a link might be allowed to carry the full amount of its requested traffic at one time and zero at another.

The second scheme involves assigning traffic on each link proportional to a weighted cost. Thus,

\[
u_{f,i} = \min \left( \frac{w_iS_{f,i}u_f}{S_{f,i}}, C_{f,i} \right),
\]

(2.69)

where \( C_{f,i} \) is the capacity associated with upstream link \( i \), \( C_f = \sum_{i=1}^{k} C_{f,i} \). Note that for equal weighting, \( C_{f,i} \) would not be needed; however, for different \( w_i \)'s, the allocated traffic on some link might exceed its \( C_{f,i} \) and \( C_{f,i} \) would be selected for that link. If this happened, part of \( u_f \) might remain unused after all links had been considered, and the distribution of the unused traffic would be necessary. That is, the upstream traffic allocation will be divided into two phases, with the first to distribute \( u_f \) among all links according to (2.69), and the second to check
for unused traffic and make a second allocation. Here, unused traffic would be
distributed to fill the link capacities sequentially according to the values of \( S_{f,i} \)'s,
the one with the smallest \( S_{f,i} \) being serviced first.

The output traffic allocation problem involves the distribution of \( u_r \) among
downstream links so as to achieve a minimal cost. Thus, the problem is to minimize

\[
F = \sum_{i=1}^{l} S_{D,i}u_{r,i}
\]

subject to

\[
u_r = \sum_{i=1}^{l} u_{r,i}
\]

and

\[
C_{r,i} \geq u_{r,i} \geq 0,
\]

where \( u_{r,i} \) is the output flow to be allocated on link \( i \), \( C_{r,i} \) is the capacity associated
with that link, \( (\sum_{i=1}^{l} C_{r,i} = C_r) \), and \( S_{D,i} \) is the cost associated with downstream
link \( i \). The latter is the sum of the link travel time, the time to clear the downstream
queues (associated with downstream node \( i \)), and the aggregated downstream cost
associated with that node. The first is a known value, the second is given by the
\( t_f \) associated with downstream node \( i \), and the last would be communicated from
the nodes downstream from that node. Note that the solution is obvious, i.e., fill
up the links sequentially in the ascending order of the corresponding \( S_{D,i} \).

Normally, if there is no congestion at all of the downstream nodes, the determined \( u_{r,i} \) would be employed. However, if \( q_1' > q_{1r} \) then \( q_1' \) would be collected
into \( q_1 \) and the downstream traffic allocation would be modified to adapt to down­
stream traffic condition. This is achieved by multiplying each \( u_{r,i} \) by a factor \( w_i' \).
The latter is determined by first solving
\[ w_i' = \begin{cases} 0.95 & \text{if } q_{1,i}' > \frac{q_i'}{l}, \\ 1 & \text{otherwise.} \end{cases} \quad (2.73) \]

followed by a normalization over all \( w_i' \)s so that \( \sum_i w_i' = l \), where \( q_{1,i}' \) is the component of \( q_1' \) associated with the downstream link \( i \). Note that the number 0.95 is selected so that network travel time would remain minimal whereas downstream congestion are accounted for. Further, if \( \sum_{i=1}^l w_i' u_{r,i} \leq u_r \), the remaining \( u_r \) would be assigned to the links in the ascending order of \( S_{D,i} \).

Finally, the aggregated downstream cost \( S_D \) would be updated by

\[
\frac{1}{S_D} = \sum_{i=1}^l \frac{1}{S_{D,i}}. \quad (2.74)
\]

2.5 Estimation of Traffic Demand and Queue Lengths

2.5.1 Estimation of Traffic Demand

The estimation of traffic demand involves using the \( r_{D,S_i}(t - d_i) \), \( i = 1, \ldots, k \) from upstream nodes to calculate \( \bar{r}_{D,i}(t) \) and thus \( \bar{r}_D(t) \). A well-known unbiased estimate for \( \bar{r}_{D,i}(t) \), when \( r_{D,i}(t) \) is constant, is given by

\[
\bar{r}_{D,i}(t) = \frac{1}{n_t} \sum_{s=1}^w r_{D,S_i}^{(s)} (t - d_i), \quad (2.75)
\]

where \( r_{D,S_i}^{(s)} (t - d_i) \) is \( r_{D,S_i} (t - d_i) \) obtained at the \( s^{th} \) sampling time, and \( n_t \) is the total number of intervals measured [63]. Since this estimator can be used only when the \( r_{D,S_i}^{(s)} (t - d_i) \) are essentially fixed, it will not be employed.

Although constant \( r_D \) was assumed in the derivation of controls, the actual traffic need not necessarily be constant. Instead, the "constant" can be either an estimate of a stochastic process or an average for a deterministic one. Here, the latter will be considered and \( r_D(t) \) will be assumed to be the sum of linear functions of time defined by

54
Thus, the problem is to estimate $\theta_{i,1}$ and $\theta_{i,0}$ on line. Note that this choice can be employed to approximate typical traffic patterns.

In general, on-line estimation for such a process can be done by using either a gradient (or projection) technique or a least-squares algorithm [64]. The former involves minimizing the difference between successive estimates by using the previous estimate, and the projection of this estimate onto a hypersurface described by the current measurement. Such an estimator is simple; however, the estimated parameters converge slowly to their true values.

The latter involves minimizing the accumulated squares of the differences between the desired and the estimated output and a recursive estimator results. Initially, the estimate would converge very rapidly; subsequently, however, the convergence would become much slower. Any one of the following modifications can be employed to resolve such a difficulty: covariance modification, a finite-window approach, and exponential weighting of measured data. The first involves presetting the covariance (or projection) matrix once its trace falls below a threshold value; however, the estimate may degrade substantially before the trace of this matrix becomes too small. As a result, this estimator would not adapt quickly to demand changes. The second involves maintaining a finite length (or window) of data by discarding old data "prior to" the window size. A related problem is that the "optimal" window size depends on the parameter dynamics. In the third approach, a data-dependent forgetting factor would be employed to modify the covariance matrix and an algorithm that would adapt quickly to demand changes results. For this reason, the last approach will be employed.
The estimator presented in the following can be found in the literature on adaptive estimation and system identification, e.g., [64] and [67]. In the present work, the ARMA (Auto-Regressive-Moving-Average) model associated with \( r_{D,S_i}(t) \) is defined as

\[
r_{D,S_i}(t) = \phi^T(t)\theta(t),
\]

and the corresponding estimator is

\[
\hat{r}_{D,S_i}(t) = \phi^T(t)\hat{\theta}(t),
\]

where

\[
\theta(t) \overset{\text{def}}{=} \begin{bmatrix} \theta_{i,1}(t) & \theta_{i,0}(t) \end{bmatrix}^T,
\]

\[
\hat{\theta}(t) \overset{\text{def}}{=} \text{estimated } \theta(t),
\]

\[
\phi(t) \overset{\text{def}}{=} \begin{bmatrix} t & 1 \end{bmatrix}^T
\]

and T denotes matrix transpose. Since \( \theta_{i,0} \) and \( \theta_{i,1} \) would be updated at discrete times, the integer \( n_s \) will denote the sampling instant; i.e., \( t = n_s\Delta t_s \), where \( \Delta t_s \) is the sampling interval. Then, the demand estimator is

\[
\hat{\theta}(n_s) = \hat{\theta}(n_s - 1) + \frac{P(n_s - 2)\phi(n_s - 1)}{f(n_s - 1) + \phi^T(n_s - 1)P(n_s - 2)\phi(n_s - 1)} \times \left[ r_{D,S_i}(n_s) - \phi^T(n_s - 1)\hat{\theta}(n_s - 1) \right],
\]

where

\[
P(n_s - 1) = \frac{P(n_s - 2)}{f(n_s - 1)} - \frac{P(n_s - 2)\phi(n_s - 1)\phi^T(n_s - 1)P(n_s - 2)}{f(n_s - 1)\left[f(n_s - 1) + \phi^T(n_s - 1)P(n_s - 2)\phi(n_s - 1)\right]},
\]

\[
f(n_s) = 1 - f_s\frac{\epsilon^2(n_s)}{1 + \phi^T(n_s - 1)P(n_s - 2)\phi(n_s - 1)},
\]

\[
\epsilon(n_s) = r_{D,S_i}(n_s) - \phi^T(n_s)\hat{\theta}(n_s),
\]

56
\[
\hat{\theta}(0) = \begin{bmatrix} 0 & r_{D,S_i}(0) \end{bmatrix}^T,
\]
\[
\phi(-1) = \begin{bmatrix} 0 & 0 \end{bmatrix}^T,
\]
\[
P(-1) = P(-2) = \text{an identity matrix},
\]
\[
f(0) = f(-1) = 1,
\]

\(P(t)\) is a \(2 \times 2\) matrix, \(f(t)\) is the forgetting factor, \((1 \geq f(t) \geq 0)\), \(f_s\) is a small constant and \(e(t)\) is the prediction error [68,69].

Upon solving for \(\hat{\theta}\), the \(d_i\)-time ahead prediction of \(r_{D,S_i}\), can be obtained and the total traffic demand becomes

\[
\hat{r}_D(n_s) = \sum_{i=1}^{k} \hat{r}_{D,S_i} \left( n_s + \frac{d_i}{\Delta t_s} \right). 
\]

Note that although \(t\) and \(n_s\) would restart from zero at the beginning of each control interval, the recursive formula given by (2.81) – (2.83) would be used continuously. That is, since traffic would be continuous, the estimator would operate with an infinite horizon and the initial conditions for \(\hat{\theta}\) and \(\phi\) would be incorporated only when the estimator was initialized.

2.5.2 Determination of queue lengths

Both \(q_1\) and \(q_2\) must be estimated at each node. Since \(q_2\) is a conceptual entity, it cannot be measured. Instead, it can be approximated by adding the "previous" \(q_2\) to its variation during the past \(\Delta t_s\) time units. Thus,

\[
q_2(n_s) = q_2(n_s - 1) + \Delta t_s \left( r_D - u_f (n_s - 1) \right). 
\]

The aggregated queue \(q_m\), which is measurable at \(N_0\), is the sum of \(q_1\) and \(q'_1\), the latter being the sum of the \(q_2\)'s associated with downstream nodes. Thus, \(q_1\) could be determined as \(q_m - q'_1\). However, since queue-length measurements tend
to fluctuate rapidly, Levine [66] suggests that such an approach may be inaccurate. Instead, he recommends that both traffic flow and measured queue sizes be employed to obviate this difficulty and a corresponding simple choice is

\[
q_1(n_s) = \frac{q_1(n_s - 1) + \Delta t_s \left( u_f(n_s - 1) - u_r(n_s - 1) \right)}{2} + [q_m(n_s) - q_1'(n_s)]. \quad (2.86)
\]

According to the protocol shown in Fig. 3, the queues employed in the controls must be \(d_i\)-time ahead predictions; thus (2.85) and (2.86) will not be used directly in determining the next controls. Instead, they will be used as "measured" quantities and the predicted values obtained from (2.80), with \(r_{D,S_i}\) replaced by \(q_1(n_s)\) and \(q_2(n_s)\) will be employed. Note that after the queues are reset, \(n_s\) will restart from \(0\), \(q_1'(0) = 0\), and \(q_1(0) = q_m(0)\).

2.6 Discussion and Conclusion

A coordinated routing scheme has been presented for a single-destination network under constant traffic-demand conditions. In the free final time case, closed-loop solutions were derived for the optimal controls. In fixed, final-time case, integral forms of controls were derived and closed-loop as well as open-loop information (initial conditions) were involved. The corresponding control domains, in terms of queue lengths and average traffic rate were also specified. Since the Hamiltonian is independent of the demand function provided both controls are unsaturated, the results from both cases are valid for both the constant- and time-varying demand cases.

Nodal queue, instead of link queue, was employed to make the problem tractable. This is justifiable if one considers the traffic heading for a destination from different upstream links, where the vehicles waiting for service can be 'lumped' together. For multi-destination case as will be discussed in Chapter 4, this is also true since
the queues for each destination interact with each other. Further, since the description of the detailed interaction could be extremely difficult, an aggregated nodal queue should be an appropriate choice.

The use of average traffic demand in the controls is a good approximation only when the queues are large. For example, in the free $t_f$ case, for $\alpha_1 = \alpha_2 = 1$, $q_1(0) = q_2(0) = 1$, and $r_D(t) = 10t + 5$, then $t_f = 3.34$, $r_D = 11.5$, and $u_f(0) = 12.5 \gg r_D(0)$ would result — an unreasonable result since $r_D(0)$ is only 5. The situation would be even worse if $q_1(0) = q_2(0) = 0$; then $t_f$ cannot be determined. This difficulty can be overcome by using the real-time demand in the controls if the queue sizes are in $\Omega_0 \cap \Omega_1 \cap \Omega_2$; and using the average demand only when saturated control(s) were necessary.

A result of the coordination process was that part of the queue could be stored at upstream nodes. Under light-to-moderate traffic conditions, these queues can be cleared; however, in heavy traffic conditions, they might become excessive. A reset mechanism was introduced to circumvent this difficulty. In the free final-time case, since the solutions were in closed-loop form, such a mechanism would not create any difficulty. In the fixed final-time case, because of the initial conditions involved in the controls, the latter might not respond to the queue resetting at upstream nodes during a control interval. To resolve this difficulty, either large $\alpha_2$’s associated with downstream nodes would be selected so that $q_1'$ would be kept small, or the queue would be reset at the end of each control interval thus making excessive $q_1'$ less probable thereafter. The former choice might reduce the efficacy of coordination since most requesting traffic would be accepted despite any downstream congestion. The latter approach could result in a step change in arrival rate to downstream nodes and thus deteriorate network performance. To avoid this, synchronous operation might be preferable. That is, in the free
final-time case, asynchronous operation would be desirable and, to preserve local optimality, a queue would not be reset unless it would become excessive; in the fixed final-time case, synchronous operation would be preferred and queues would be reset at the end of each control interval.

A scheme was also proposed when queues could not be cleared. In this case, if the downstream links were not capable of carrying the requested traffic, input traffic would be restricted to avoid serious congestion. If the upstream links could not handle the sum of the requested traffic and the equivalent flow due to \( q_2 \) (e.g., see equation (2.12c)), the input traffic would be saturated and an optimal solution in the sense of clearing \( q_1 \) while minimizing \( (C_f - u_r)^2 \) was derived.

A closed-loop link might result when the downstream traffic allocation scheme is applied to a cyclic network, i.e., a network with one or more loops. This property, however, is shared by most decentralized control algorithms with completely aggregated information about other nodes. From an operational point of view, this problem can be removed by either keeping a record in each vehicle of the nodes it has traversed or assigning a link as inadmissible with respect to the destination.

While traffic rates and queue lengths can be measured, the measurements might involve noise, and changes in queue lengths might be too fast for a measurement to be useful; thus, it may be necessary to estimate these quantities. A least-square scheme with a forgetting factor was specified for traffic rate estimation and a scheme that combines both traffic flow and queue size measurements was proposed for the computation of queue lengths.

It is hypothesized that the coordinated traffic control scheme described here will result in smooth traffic flow, reduced fuel consumption, and near-minimal aggregate travel times. The simulation study, which is reported in Chapter 3, is intended to evaluate this hypothesis.
3.1 Introduction

The control laws derived in Chapter 2 are evaluated by considering a simple network. Here, the four-node acyclic network shown in Fig. 19 is employed and simulation results are obtained. In this network, node 1 is a source node, nodes 2 and 3 are internal as well as source nodes, and node 4 is a destination node.

The delay-free, link travel times \( t(i, j) \), in minutes, from node \( i \) to node \( j \) are: \( t(1, 2) = t(3, 4) = 3, t(1, 3) = 4, t(1, 4) = t(2, 3) = 5, t(2, 4) = 6 \), and the flow-rate capacities \( Cap(i, j) \), in vehicles/minute, associated with the link from node \( i \) to \( j \) are: \( Cap(1, 2) = Cap(1, 3) = Cap(1, 4) = 50, Cap(2, 3) = Cap(2, 4) = 30 \), and \( Cap(3, 4) = 70 \). The numbers are selected as being characteristic of an automated urban highway network with a 1-second headway policy [74]. Under this policy, 60 veh/minute per lane is theoretically possible; however, due to the number of lanes and the complexity of roadway geometries, such as the number of entry/merge points, this number may be less, and 50 veh/minute is selected for a one-lane roadway characterized by relatively little traffic interaction, 30 veh/minute is for one with many entry/merge points, and 70 veh/minute is for a two-lane roadway.

This network is considered to be representative as it encompasses a variety of possible traffic situations. At node 1, vehicles enter the network and are dispatched onto three links with different costs; at node 2, traffic diverges onto two links; at
node 3, traffic merges from two upstream links; and at node 4, traffic merges from three links then leaves the network. Further, at node 2, the aggregated upstream capacity is less than the aggregated downstream capacity, whereas at node 3, the reverse is true. Note that the latter makes it possible to study the effect resulting from the imbalance between input and output capacities.

Since the purpose of the simulation study is to study the efficacy of the proposed control algorithm in handling traffic, both single external-demand and three external-demand situations are considered. In the single-demand case, the variation of the travel time with respect to the demand entering the network at node 1 is explored, together with the ability of the control algorithm to achieve queue reduction, adapt to time-varying traffic, and to service rush-hour traffic. Finally, the performance of the network when limitations exist at the exit is studied. In the multiple external-demand cases, the factors enumerated above are evaluated.
under both moderate- and heavy-demand conditions.

Although the control laws are in continuous-time, the use of a computer dictates a discrete-time implementation, and the sampling interval $\Delta t_s$ must be determined. Normally, this quantity must be 6 times less than the smallest time constant of the system under control. Here, from equation (2.12a), the time constant associated with each node is $\frac{1}{a_1} = \sqrt{a_4} S_D$. With $a_4 = 1$ at every node$^1$, this value becomes $S_D = 1 / \sum_{i=1}^{\infty} \frac{1}{S_{D,i}}$. For the specified network, $S_D = 2, 3.43,$ and 3 at nodes 1, 2, and 3, respectively; thus, $\Delta t_s$ must be less than 0.33 and $\Delta t_s = 0.25 \text{min.}$ is selected. At the destination node, if the egress capacity is not limited, the time constant associated with clearing the queues is assumed to be 6 sampling intervals. If the egress capacity is limited, this number is assumed to be 60 sampling intervals instead of 6. This large increase is necessary to reflect a congested destination node since there is no downstream node. Since the main effect of this number is the rate at which $q_1$ changes, a value that is 10 times that of the unlimited-capacity case is considered to be appropriate.

The $q_{1r}$ beyond which $q_1$ would be collected into $q_1$ is selected as 10. This choice is a compromise between optimality and adaptivity — the former requiring a large $q_1$ and the latter a small one. Ideally, an unconstrained $q_1$ would be desirable since this is the condition under which the optimal controls were derived; however, from a practical point of view, a large $q_1$ corresponds to an unaccounted for congestion which, when added to $q_1$, might result in a congestion that exceeds the node storage capacity. Further, the larger the $q_{1r}$, the less probable that the reset mechanism would be activated. This would be undesirable since the downstream traffic allocation might not adapt quickly enough to changing traffic conditions.

$^1$Other numbers are possible; however, the principal effect will be the rate at which $q_1$ changes, and 1 is chosen for convenience.
A network simulation program was developed and the associated flow chart is included in Appendix A. The interested reader may consult the author for the program listing.

In the results subsequently presented, the predicted values of both the traffic arrival rate and the $q_2$ at each node are shown as functions of time when vehicles would arrive at the corresponding node. Further, time refers to sampling instant unless otherwise specified, and the network is said to be in a rush-hour condition if the input demand is equal to or greater than the minimum of the upstream and downstream capacities at any node in the network.

### 3.2 Single External-Demand Situation

An external demand to node 1 is assumed and both constant and time-varying traffic are considered. Four cases are studied: 1) constant demand without any initial queues; 2) constant, moderate demand with initial queues; 3) a sinusoidally changing demand; and 4) a limited exit-capacity situation. Note that 3) is intended to model the change from moderate to rush-hour traffic.

**1) Constant-demand case**

The response of the network to various constant values of external demand is evaluated. It is instructive to consider two cases, a moderate demand of 90 veh./min. and an extreme demand of 152 veh./min. in detail and then examine summary results for other cases.

The results associated with the former are shown in Fig. 20. Here, flow rates are plotted versus time, and data associated with each node are presented. First note that both the input and the output traffic through node 1 is a constant value, whereas that through node 2, 3, and 4, while initially zero, subsequently increases. The traffic through nodes 2 and 4 approaches a constant value whereas that at
node 3 decreases to zero. The reasons for this are clear from an evaluation of the link travel times. Further, since the traffic is moderate, the node output traffic tracks the input at every node, and all queues remain zero. Finally, it is determined that the average travel time from node 1 to node 4 is 6.68 minutes.

The results for a constant external demand of 152 veh./min. are shown in Fig. 21 and 22. In the former, where queues are shown as functions of time, it is observed that the $q_1$ at both node 1 and node 3 increase monotonically to $q_{1s} = 50$ as expected whereas the $q_2$ at node 1 increases linearly. Fig. 22 indicates that although the external demand is heavy, the traffic is smooth at each node. Finally, the average travel time is found to be 39.13 minutes — a very large value which corresponds to a congested network. Clearly, these results are unacceptable — a queue which increases linearly without bound and an average travel time which is some 6 times that for the delay-free case.

Next consider the network performance for various constant demands between 45 and 152. The corresponding average travel time is plotted versus demand as shown in Fig. 23. Note that travel time increases substantially after the demand reaches 150, which is the network capacity. As the demand further increases, the travel time “soars” as a result of congestion. Clearly, the external demand should be limited to the network capacity if excessive delays are to be avoided.

2) Constant demand with initial queues

The efficacy of the proposed control algorithms in clearing queues is investigated. Since it is desired to perform this study with the network in a steady-state condition, the results obtained during a transient initialization period are discarded. Here, this period is defined to be a time interval after which both the

---

2 This travel time was obtained by averaging over run time. If the run time were longer, this delay would be longer — approaching infinity as run time became large.
Figure 20: Traffic under constant, moderate demand
Node 1 input = 152 veh/min
- for q1
x for q2

Node 2

Node 3

Node 4

Figure 21: Queues under constant, heavy demand
Figure 22: Traffic under constant, heavy demand
Figure 23: Travel time as a function of demand

traffic and the estimators stabilize; then the local clock at each node is restarted from 0. From Fig. 20 it was found that this period was approximately 90, and 100 is selected for the current case.

A moderate demand of 60 veh./min. and an initial \( q_1 \) of 50 vehicles at every node is selected as a representative case. The resulting queue-length histories and traffic rates as functions of time are shown in Fig. 24 and 25, respectively. Note from Fig. 24 that the \( q_1 \)'s decrease in a near-exponential fashion and from Fig. 25 that the output traffic converges smoothly to the input traffic except at node 4. This is because node 4 is the destination where unlimited egress capacity and a time constant of 6 were assumed. Finally, the average travel time from node 1 to node 4 is found to be 5.89 minutes — a value larger than 5.76 minutes for \( q_1 = 0 \) case as can be obtained from Fig. 23. Thus, under moderate demand conditions, queues can be cleared. Further, since more traffic would result with initial queues,
the average travel time would become longer as part of the traffic would take alternate routes.

3) Sinusoidally-changing demand case

The ability of the network to handle a time-varying external demand is investigated by considering several such demands. The first is a moderate choice of $r_D(t) = 90 + 20 \sin(0.05t)$ veh./min. at node 1, with all queues initially zero. The choice of the angular frequency 0.05 corresponds to a cycle of approximately 125.7 minutes or 503 sampling intervals, i.e., the positive half of this cycle is approximately 1 hour and can be employed to simulate rush-hour traffic. The results are shown in Fig. 26 from which it is noted that both the input and the output node traffic “track” the demand and, as a result, the queues remain zero and no congestion is present. The travel time is determined as 6.68 minutes — essentially the expected result since no congestion is present.

The second is a heavy demand of $r_D(t) = 130 + 30 \sin(0.05t)$, which again corresponds to a rush hour of approximately one hour. Here again, all initial queues are zero. The results are shown in Figs. 27 and 28. The former shows that queues increase at the beginning of the rush hour and decrease at the end; whereas the latter indicates that the difference between the input and the output traffic is small even during the rush hour period. Finally, the average travel time is determined as 21.09 minutes — approximately 3 times that for the moderate-demand case.

Next consider the demand $r_D(t) = 120 + 40 \sin(0.05t)$ with initial queues zero. The results are similar to those of the previous case with an average travel time of 18.65 minutes. Obviously, although the peak traffic demands between these cases are the same, the difference in travel time result from the different average demands, the larger this demand, the more serious the congestion and the longer the travel time. That is, the longer the rush hour or the higher the peak demand,
Figure 24: Queues under constant, moderate demand with initial queues
Figure 25: Traffic under constant, moderate demand with initial queues
Node 1 input = 90 + 20sin(0.05t)(veh/min)
- for input traffic
x for output traffic

<table>
<thead>
<tr>
<th>Time (min)</th>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>252.00</td>
<td>252.00</td>
<td>0.00</td>
<td>420.00</td>
</tr>
<tr>
<td>84.00</td>
<td>336.00</td>
<td>336.00</td>
<td>84.00</td>
<td>504.00</td>
</tr>
<tr>
<td>168.00</td>
<td>420.00</td>
<td>420.00</td>
<td>168.00</td>
<td>252.00</td>
</tr>
<tr>
<td>252.00</td>
<td>504.00</td>
<td>504.00</td>
<td>252.00</td>
<td>336.00</td>
</tr>
<tr>
<td>336.00</td>
<td>84.00</td>
<td>84.00</td>
<td>336.00</td>
<td>420.00</td>
</tr>
<tr>
<td>420.00</td>
<td>168.00</td>
<td>168.00</td>
<td>420.00</td>
<td>252.00</td>
</tr>
<tr>
<td>504.00</td>
<td>252.00</td>
<td>252.00</td>
<td>504.00</td>
<td>336.00</td>
</tr>
</tbody>
</table>

Figure 26: Traffic under sinusoidally changing, moderate demand
the longer the average travel time.

4) Limited egress-capacity case

The effect of limited exit capacity on the proposed control algorithm is studied. Here, with the input capacity of 150 veh./min., an output capacity of 100 veh./min. at the destination node is considered. Further, vehicles are assumed to enter the network at node 1 and a demand of $r_D(t) = 95 + 10\sin(0.05t)$ veh./min. is employed.

The results are shown in Fig. 29 and Fig. 30. The former shows that at node 4, $q_2$ increases at time 55 as a result of estimated demand from node 2. Thus, the component of this $q_2$ stored there becomes larger than $q_1$ and the reset mechanism is activated. As a result, $q_1$ at node 2 "jumps". Subsequently during the rush hour, a similar situation happens at nodes 1, 2, and 3. That is, by resetting the queues, congestion is "pushed" toward the upstream nodes. Finally, the travel time is determined as 10.26 minutes.

Thus, with the proposed control algorithm applied to a network in which the egress capacity is limited, congestion would form before the exit and propagate to upstream nodes during a rush hour. At the end of the rush hour, these queues would be cleared; however, because of the congestion, travel time through the network would be longer. Further, with larger demands or non-zero initial queues, congestion would be more serious at every node and travel time would be longer.

3.3 Three External-Demand Situation

A network involving multiple-input traffic is investigated and network performance characteristics such as traffic flow, congestion, and travel time are studied. The network of Fig. 19 with external demands entering at node 1, 2, and 3 is employed, and both moderate and heavy external demands are considered.
Figure 27: Queues under sinusoidally changing, heavy demand
Figure 28: Traffic under sinusoidally changing, heavy demand
Figure 29: Queues under moderate demand, with exit congestion
Node 1 Input = 95 + 10sin(0.05)t (veh/min); Cexit = 100
- for input traffic
x for output traffic

Figure 30: Traffic under moderate demand, with exit congestion
1) Moderate demands

Since under moderate traffic conditions, the results from both constant and
time-varying demand situations are similar, only the latter are considered here.
These demands are selected so that they would not saturated any link capacity in
the network. Here, \(30 + 10 \sin(0.05t)\), \(20 + 10 \sin(0.05t)\), and \(10 + 10 \sin(0.05t)\),
at nodes 1, 2, and 3, respectively are employed with the results shown in Fig. 31.
Note that smooth traffic flow through the network is achieved. Further, since the
output traffic “tracks” the input at each node, the queues remain at zero. Finally,
the average travel times to node 4 are determined as 5.05 minutes, 6.09 minutes,
and 3.08 minutes, from node 1, 2, and 3, respectively. Note that the travel time
from node 1 to the destination is the same as that for the single-external demand
case as can be seen in Fig. 23. Generally, it appears that, under moderate traffic
conditions, results from the multiple-demand case should be similar to those from
the single-demand case.

2) Heavy demands — No Exit Restrictions

The efficacy of the proposed control algorithms in dealing with strong inter­
actions among vehicles from different entry points is investigated. Here, \(r_D(t) =
40 + 12 \sin(0.05t)\) at nodes 1, 2, and 3 are selected and congestion should result.
The results are shown in Fig. 32 and 33. Note from the former that the \(q_1\)s at
nodes 1, 2, and 3 increase as a result of resetting during the rush hour when the
input traffic at each node exceeds the output capacity. Further, the \(q_2\)s associated
with external demands at nodes 2 and 3 increase during this period. From Fig. 33,
it is observed that the output traffic increases after \(q_1^'\) is collected into \(q_1\). Finally,
the average travel times to node 4 are found to be 7.60 minutes, 11.15 minutes,
16.45 minutes from nodes 1, 2 and 3, respectively.
Figure 31: Traffic under 3 moderate demands
In summary, like the heavy traffic situation in the single-demand case, congestion resulting from heavy demand at different inputs would be “pushed” toward the entry points where vehicles would be restricted from entering the network. Further, as a result of downstream congestion, the travel time from node 1 is larger than in the single-demand case. At the end of the rush hour, congestion would be cleared. If the demands were in-phase spatially, the peak value of congestion would be larger; however, the congestion would probably be cleared more quickly. Further, with greater external demands at each node, travel times would become longer.

3.4 Conclusion and Discussion

The control algorithms developed in Chapter 2 were employed in a network comprised of four nodes. Both single external-demand and three external-demand situations were studied. In the former, both constant- and time-varying demands were considered. In the constant-demand case, travel times were determined for both moderate- and heavy-demand situations and the relationship between average travel time and demand was plotted. According to this curve, travel time increased dramatically as demand exceeded the network capacity. In the time-varying demand case, both moderate and heavy traffic situations were studied and the results indicated that the controls “adapted” to changing demand very well. Further, in the moderate traffic situation, since no congestion were present, the results in both the constant demand and the time-varying demand cases were similar.

The efficacy of the algorithm in clearing queues was also considered. The results showed that queues were cleared in a near-exponential fashion. This is desirable as it approximates actual traffic flow closely[70].

Limited egress capacity situation was explored. The results showed that,
Figure 32: Queues under 3 heavy demands
Figure 33: Traffic under 3 heavy demands
with the resetting mechanism, congestion were pushed in the upstream direction. Further, congestion was distributed at different upstream nodes and congestion would not tend to concentrate at one node.

In the three external-demand case, time-varying traffic were considered. If the demands were moderate, the dynamic of queues and traffic were similar to those in the single-demand case; if the demands were heavy, queues increased at the beginning of the rush hour and decreased at the end. The result associated with the latter also indicated that congestion propagated in the upstream direction and vehicles would be restricted from entering the network at the entry points.

This simulation study showed that under the framework proposed in Chapter 2 for single-destination flow and routing control, upstream and downstream nodes would coordinate and adapt to changing traffic situations given no a priori information about traffic conditions so that smooth traffic flow and near minimal travel time would result. Nevertheless, such a single-destination case is too restricted as multiple destinations are usually involved in an actual network. Thus, the flow and routing control of such a network will be addressed in the next chapter.
CHAPTER IV
MULTI-DESTINATION ROUTING

4.1 Introduction

In practice, most traffic networks involve both a number of nodes at which traffic may enter and a number of nodes at which it may depart; thus it is necessary to evaluate multi-destination routing. In previous studies, a centralized computation structure was employed and routing was achieved (or to be achieved) by incorporating a system surveillance capability to update route guidance parameters in roadside units [26,27,28]. In theoretical studies, routing was formulated either as a nonlinear, multi-commodity flow problem [40,42,43,63], or as an optimal control problem [16,21,34,35]. The former involves a cost function based on a queueing model, a computationally intensive, iterative procedure and an approach only useful for steady-state analysis, while the latter involves either a centralized control structure or a departure bay for each destination. Further, both formulations dealt with routing independently of flow control. Since the interest here is in designing a coordinated decentralized scheme, which incorporates routing and flow control, a method similar to the single-destination routing algorithm of Chapter 2 is proposed.
4.2 Problem Formulation and Solution

Two approaches were considered in the multi-destination routing problem. The first involved solving for the optimal flows at a node and then distributing the output traffic based on the cost associated with each connecting downstream link. This involves maximizing the traffic on the lower-cost links to each destination subject to various constraints. However, this could cause "far-end" traffic (i.e., traffic heading for a destination far away from the node of interest) to wait an excessive amount of time and, due to the sequential nature of traffic flow, to block "near-end" traffic (i.e., traffic heading for a destination close to the node of interest) thus creating unnecessary congestion.

The second approach, which involved the inclusion of constraints to ensure "fairness" among various destinations, has three hierarchical levels. The first corresponds to the determination of the optimal flow rates both into and out of a node, the second involves the distribution of these rates to different destinations so as to achieve smooth flow, and the third relates to the allocation of this traffic onto different links for each destination. Since the first is conceptually similar to the single-destination case, the analysis will not be repeated here.

The second level involves both upstream and downstream traffic distribution for different destinations. Here, for the sake of fairness, a proportional scheme is proposed for the upstream traffic allocation and

\[ u_f^j = \begin{cases} \frac{r_D^j}{r_D} u_f & \text{if } r_D^j \neq 0, \\ \alpha_2 q_2^j & \text{if } r_D^j = 0 \end{cases} \]  

results, where \( u_f^j = \sum_{i=1}^{k} u_{f,i}^j \) and \( r_D^j = \sum_{i=1}^{k} r_{D,i}^j \) are the aggregated upstream traffic and demand for destination \( j \), respectively, \( u_{f,i}^j \) and \( r_{D,i}^j \) are the destination \( j \) traffic and demand on upstream link \( i \), respectively, \( q_2^j \) is the component of \( q_2 \) for
destination \(j\), and \(k\) is the number of upstream links. Since, from equation (2.12c),
\[ u_f = r_D + \alpha_2 q_2, \]
a flow due to \(q_2\) is included if \(r_{D,i}^j \neq 0\). Further,
\[ u_{f,i}^j = \begin{cases} r_{D,i}^j u_f^j & \text{if } r_{D,i}^j \neq 0, \\ \frac{r_{D,i}^j}{r_D} u_f^j & \text{if } r_{D,i}^j = 0, \end{cases} \tag{4.2} \]
where \(q_{2,i}^j\) is the component of \(q_2\) on upstream link \(i\) with destination \(j\).

Two approaches were considered for the determination of the downstream traffic to each destination, \(u_r^j\). The first involved minimizing the sum of the squares of the differences between \(u_r^j\) and \(u_f^j\) given \(u_r\), \(u_f\), \(u_r^j\) and the positivity constraints. The solution was
\[ u_r^j = u_f^j + \frac{u_r - u_f}{n} \]
where \(n\) is the number of destinations. In an unbalanced-demand situation (i.e., the volumes of traffic heading for different destinations are not equal) the \(u_r^j\)'s might not be the best choice. For example, in a two-destination network with \(u_1^j = u_f^j = 0\), \(u_1^j = \frac{u_r + u_f}{2}\) and \(u_2^j = \frac{u_r - u_f}{2}\), and the latter is not necessarily zero as would be desired.

The second approach was to assign \(u_r^j\) in proportion to \(u_f^j\) and
\[ u_r^j = \begin{cases} \frac{u_r}{u_f} u_f^j & \text{if } u_f^j \neq 0, \\ \alpha_1 q_1^j & \text{if } u_f^j = 0. \end{cases} \tag{4.3} \]
results, where \(q_1^j\) is the component of \(q_1\) for destination \(j\). Note that \(u_r^j\) would be zero if \(u_f^j = q_1^j = 0\). Further, the equivalent flow due to \(q_1\) would be included in \(u_r\) if \(u_f^j \neq 0\) since, from equation (2.12d), \(u_r = \alpha_1 q_1 + u_f\).

Similar to the single-destination case, the objective in the determination of \(u_{r,i}^j\) is to minimize
\[ F_2 = \sum_{i=1}^{l} \sum_{j=1}^{n} S_{D,i}^j u_{r,i}^j, \tag{4.4} \]
subject to

\[ u_r^j = \sum_{i=1}^{l} u_{r,i}^j, \tag{4.5} \]

and

\[ \sum_{j=1}^{n} u_{r,i}^j \leq C_{r,i}. \tag{4.6} \]

where \( S_{r,i}^j \) is the downstream cost associated with link \( i \) for destination \( j \) traffic, and \( C_{r,i} \) is the capacity of downstream link \( i \). Since this is a linear programming (LP) problem, well-known solution techniques, such as the simplex method, can be employed [73] to solve for \( u_{r,i}^j \). Note that if \( n = 1 \), the problem reduces to the single-destination case and \( C_{r,i} \) would be filled sequentially in the ascending order of \( S_{D,i}^1 \). Further, the third-level computation would not be necessary if \( l = 1 \) since the \( u_{r,i}^j \)'s are uniquely determined by \( u_r^j \).

In general, the solution in LP is to maximize the \( u_{r,i}^j \)'s associated with the minimal \( S_{D,i}^j \)'s. If two or more \( S_{D,i}^j \)'s are equal, the resulting \( u_{r,i}^j \)'s might not be equal and traffic could concentrate on one link thus increasing the probability of congestion. Thus, a procedure that would even out the \( u_{r,i}^j \)'s associated with the equal or approximately equal \( S_{D,i}^j \)'s was incorporated.

In a multi-destination network, a loop might exist and, if so, it would be necessary to ensure that vehicles are not routed back to their starting points. Here, a link is assigned as admissible with respect to a destination if the traffic to that destination would not result in a loop; otherwise, it is assigned as inadmissible and no traffic for the corresponding destination would be routed onto this link.

It is assumed that only one departure lane (or bay) is associated with each node as this is a situation commonly encountered in current practice. If demand levels were light and the queues were zero, vehicles would travel to their destinations in the sum of the delay-free, link travel times. However, if demand levels were
heavy, traffic would interact at each node and delays would result. The delays due to queue formation can be accounted for as was done in Chapter 3; however, those due to vehicle maneuvering and thus travel at reduced speeds and the use of a single departure bay per node pose special problems because of their probabilistic nature. Here, it was decided to employ a fixed cost per node, or hop, to account for such factors when demand levels are heavy and queues are zero\(^1\). The greater the number of nodes, or “hops”, that a vehicle must traverse, the greater this delay; thus, the “hop count” cost, which is a function of the number of nodes along a path, must be included. Its inclusion could improve network performance in heavy-demand situations since some traffic interaction would be prevented; however, in light-demand situations the shortest path might become the most “expensive” path after the hop-count cost is included. Recall in Chapter 2 that \(t_f\) was assigned a fixed number if queues were initially zero and it would be convenient to employ this number as the cost associated with a “hop”. Normally, this number should be kept small to avoid distorting the total cost; however, since too small a \(t_f\) would result in an oscillatory estimate, a moderate value would be more appropriate\(^2\). Since the inclusion of the hop-count cost might deteriorate network performance under light-demand conditions and improve it under moderate- to heavy-demand conditions, this cost would not be included in the former case.

Basically, the decision on the threshold demand level beyond which the hop count would be included depends on the network topology and the input imbalance;

\(^1\)If queues are not zero, the computed \(t_f\), which is the time to clear congestion, would be employed as the cost of the corresponding “hop” and it is not necessary to employ such a fixed cost.

\(^2\)For the network to be studied in the next section, this number is chosen as 10 sampling intervals. For this particular network, this number was found to be a “good” choice; however, for a general network, this number must be determined through a simulation run.
Figure 34: A highway system in a typical in-land city

i.e., the imbalance in traffic distribution to all destinations. If a node along a path is located on the unique path between another source-destination (S-D) pair, or if the demand-imbalance is serious and the excessive demands converged to the same node, the threshold value must be reduced. In general, simulation runs must be conducted to determine the "best" threshold choice.

4.3 Simulation Studies

The effectiveness of the proposed solution is evaluated by means of a detailed simulation study in which the network of Fig. 34, which was first considered by Rule [33] for the fixed-route case, is employed. This network is an abstraction of the current interstate highway system of Columbus, Ohio, as well as for other large in-land cities such as Indianapolis, Indiana, or Nashville, Tennessee. This network is comprised of 8 nodes of which nodes 1 — 4 are considered as both source and destination nodes while nodes 5 — 8 are considered as internal nodes.
Normally, each node in the inner loop would have an entry and/or an exit associated with it; however, to make simulations more manageable, they are omitted\(^3\). Nevertheless, they are assumed to be present in the determination of \(\Delta t_s\) as traffic interaction would be present. Here, since the smallest delay-free link travel time in the network is 2, the smallest time constant in the network is approximately \(S_D \approx 2\); thus \(\Delta t_s = 0.25 < 0.333\) is selected.

The delay-free link travel times in minutes are shown in this figure with those times being obtained from measured distances and an assumed traffic speed of 60 mph. All links are assumed to have a capacity of 50 veh/min, a choice which is based on a 1-second time-headway policy and a single-lane of traffic\(^4\).

In order to avoid vehicles from being routed back to their starting point, the outgoing links at each node are assigned as inadmissible with respect to some destinations, as shown in Fig.35. Here, the numbers next to each link at a node correspond to the destinations to which the link is inadmissible. Consider, for example, the situation at node 1 which is highlighted in Fig. 36. The numbers 2,3,4 to the left of the node represent the destinations with respect to which the link going to the left is inadmissible. Similarly, the numbers on the top, the bottom, and to the right are for the destinations with respect to which these links are inadmissible.

Since the purpose of the simulation study is to study the efficacy of the proposed algorithm in handling traffic, both balanced and unbalanced demand sit-

\(^3\)It would not be conceptually difficult, but extremely time-consuming to expand the simulation to treat all nodes as both source and destination nodes

\(^4\)Currently, Columbus highway system is characterized by up to 4 traffic lanes in each direction. The choice made in this study corresponds to the assumption that only 1 lane would be automated.
Note: The numbers correspond to the destinations associated with which the output link is inadmissible.

Figure 35: Destinations associated with which the downstream links are inadmissible.

Figure 36: Inadmissible destinations on each link at Node 1.
uations are considered in the simulation study. The first involves equal external demands at each node and a uniform distribution to each destination. In this case, the variation of travel times with respect to the demands entering the network is explored, together with the ability of the control algorithm to achieve queue reduction, and to service rush-hour traffic. Further, the effect of stochastic demands together with the network performance when limitations exist at the exit is studied. The second involves equal external demands but a nonuniform distribution to each destination. The variation of travel times with respect to external demands under various nonuniform-distribution situations is explored.

The same simulation programs for single-destination control were employed and the associated flow charts are included in the Appendix.

4.3.1 Balanced-Demand Situation

Five cases are studied: 1) constant demands without initial queues; 2) constant demands with initial queues; 3) time-varying demands; 4) stochastic demands; and 5) a restricted egress-capacity situation.

1) Constant demand without initial queues

The response of the network to both moderate- and heavy-demand situations are investigated, and the relationship between travel time and demand level is determined. Since the demands are constant and balanced, 135 veh/min was selected as the threshold beyond which the hop count would be included. In the moderate demand case, $r_D = 50$ veh/min is employed and the resulting traffic at nodes 1 through 8 are shown in Fig. 37 and 38. Note that the input and output traffic through nodes 1, 2, 3, and 4 is initially 50 and subsequently increases to 100, which is the sum of the traffic entering from outside the network and that arriving from other nodes. Further, the traffic through nodes 5, 6, 7, and 8, while
<table>
<thead>
<tr>
<th>Node</th>
<th>Input Traffic (veh/min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00, 84.00, 168.00, 252.00, 336.00, 420.00, 504.00, 0.00</td>
</tr>
<tr>
<td>2</td>
<td>0.00, 84.00, 168.00, 252.00, 336.00, 420.00, 504.00, 0.00</td>
</tr>
<tr>
<td>3</td>
<td>0.00, 84.00, 168.00, 252.00, 336.00, 420.00, 504.00, 0.00</td>
</tr>
<tr>
<td>4</td>
<td>0.00, 84.00, 168.00, 252.00, 336.00, 420.00, 504.00, 0.00</td>
</tr>
</tbody>
</table>

Figure 37: Traffic under moderate demand — nodes 1, 2, 3, and 4
Figure 38: Traffic under moderate demand — nodes 5, 6, 7, and 8
Table 1: Travel times with constant, balanced demands

<table>
<thead>
<tr>
<th>S-D pair</th>
<th>External Demand (veh/min/node)</th>
<th>50</th>
<th>60</th>
<th>80</th>
<th>120</th>
<th>143</th>
<th>143.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2</td>
<td></td>
<td>12.07</td>
<td>12.54</td>
<td>13.13</td>
<td>13.75</td>
<td>14.48</td>
<td>16.7</td>
</tr>
<tr>
<td>1,3</td>
<td></td>
<td>14.03</td>
<td>14.09</td>
<td>14.03</td>
<td>14.10</td>
<td>14.72</td>
<td>33.69</td>
</tr>
<tr>
<td>1,4</td>
<td></td>
<td>12.07</td>
<td>12.54</td>
<td>13.13</td>
<td>13.75</td>
<td>14.52</td>
<td>17.0</td>
</tr>
<tr>
<td>2,1</td>
<td></td>
<td>12.07</td>
<td>12.54</td>
<td>13.13</td>
<td>13.75</td>
<td>14.48</td>
<td>15.32</td>
</tr>
<tr>
<td>2,2</td>
<td></td>
<td>12.07</td>
<td>12.54</td>
<td>13.13</td>
<td>13.75</td>
<td>14.52</td>
<td>16.25</td>
</tr>
<tr>
<td>2,3</td>
<td></td>
<td>14.03</td>
<td>14.09</td>
<td>14.03</td>
<td>14.10</td>
<td>14.71</td>
<td>31.37</td>
</tr>
<tr>
<td>2,4</td>
<td></td>
<td>14.03</td>
<td>14.09</td>
<td>14.03</td>
<td>14.10</td>
<td>14.71</td>
<td>31.37</td>
</tr>
<tr>
<td>3,1</td>
<td></td>
<td>14.03</td>
<td>14.09</td>
<td>14.03</td>
<td>14.10</td>
<td>14.75</td>
<td>33.28</td>
</tr>
<tr>
<td>3,2</td>
<td></td>
<td>12.07</td>
<td>12.54</td>
<td>13.13</td>
<td>13.75</td>
<td>14.56</td>
<td>16.94</td>
</tr>
<tr>
<td>3,3</td>
<td></td>
<td>12.07</td>
<td>12.54</td>
<td>13.13</td>
<td>13.75</td>
<td>14.6</td>
<td>17.14</td>
</tr>
<tr>
<td>4,1</td>
<td></td>
<td>12.07</td>
<td>12.54</td>
<td>13.13</td>
<td>13.75</td>
<td>14.57</td>
<td>15.79</td>
</tr>
<tr>
<td>4,2</td>
<td></td>
<td>14.03</td>
<td>14.09</td>
<td>14.03</td>
<td>14.10</td>
<td>14.74</td>
<td>31.22</td>
</tr>
<tr>
<td>4,3</td>
<td></td>
<td>12.07</td>
<td>12.54</td>
<td>13.13</td>
<td>13.75</td>
<td>14.6</td>
<td>16.77</td>
</tr>
</tbody>
</table>

Initially zero, approaches a constant value. Since the demands are moderate, the node output traffic "tracks" the input at every node, and all queues remain zero. Finally, the travel times between each S-D pair are shown in Table 1. Note that at this demand level, the travel time between any two nodes is approximately equal to the smallest delay-free travel time between them.

A constant demand of 143.5 veh/min at each node is employed in the heavy-demand situation and the resulting queues and traffic are shown in Fig. 39 — 40, and 41 — 42, respectively. From Fig. 39 and 40 it is observed that the queues increase dramatically with time. Ideally, the network should be able to carry demands of up to its physical capacity, which is 150 veh/min, or, the smallest sum of all link capacities at a node in the network; however, since links are assigned as inadmissible to some destinations, and extrapolation errors are involved, the actual network capacity is reduced and the demand of 143.5 veh/min is considered to have
Figure 39: Queues at nodes 1 through 4 under heavy demand levels
Figure 40: Queues at nodes 5 through 8 under heavy demand levels
Figure 41: Traffic at nodes 1 through 4 under heavy demand levels

All inputs = 143.5 (veh/min)
- for input traffic
x for output traffic

Node 1

Node 2

Node 3

Node 4

Figure 41: Traffic at nodes 1 through 4 under heavy demand levels
Figure 42: Traffic at nodes 5 through 8 under heavy demand levels
exceeded the capacity that the network can handle, which is approximately 92% of the physical network capacity. This becomes obvious by comparing the travel times for $r_D = 143.5$ veh/min with those for $r_D = 143$ veh/min — both are shown in Table 1. Note from Fig. 41 and 42 that, as a result of large queues, the output traffic starts to exceed the input.

The results for other demands are also shown in Table 1. Note that travel time from node 1 to node 2 increases with demand since more vehicles are routed onto the link with longer travel time which is the direct link connecting them. A similar increase occurs for other S-D pairs that are connected by direct links. Since travel times increase rapidly as demands approach 143.5 veh/min, this demand should be a limit imposed on network operations. Further, since the finite travel times shown here are obtained with a finite simulation period, these times would increase in heavy-demand situations if this period were extended.

2) Constant demands with initial queues

The efficacy of the proposed algorithm in clearing queues was evaluated by considering a moderate demand of 40 veh/min, $q_1(0) = 50$ vehicles at each node, and a threshold demand level of 135 veh/min. From Fig. 37 and 38, it was found that the network stabilized after 100 sampling intervals. Since the purpose in this case is to study the proposed algorithm in clearing queues, the data obtained during this transient period were discarded. At the end of this period, the clocks were reset, and the simulation restarted with $q_1(0) = 50$. The resulting queues and traffic are shown in Fig. 43 — 44, and 45 — 46, respectively. Essentially, these same results should be obtained for larger demands or initial queues if the total flow due to demands and queue sizes did not exceed the network capacity\(^5\). Note

\(^5\)The effect of queue evacuation when the total flow exceeds the network capacity can be observed in the time-varying case where the reduction of queues after a rush-hour will be studied.
Figure 43: Queues at nodes 1 to 4 under moderate demands, $q_1(0) = 50$
Figure 44: Queues at nodes 5 to 8 under moderate demands, $q_1(0) = 50$
Figure 45: Traffic at nodes 1 to 4 under moderate demands, \( q_1(0) = 50 \)
Figure 46: Traffic at nodes 5 to 8 under moderate demands, $q_1(0) = 50$
Table 2: Travel times with moderate demands and initial queues

<table>
<thead>
<tr>
<th>S-D pair</th>
<th>Travel time</th>
<th>S-D pair</th>
<th>Travel time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2</td>
<td>14.45</td>
<td>3,1</td>
<td>17.15</td>
</tr>
<tr>
<td>1,3</td>
<td>17.15</td>
<td>3,2</td>
<td>14.45</td>
</tr>
<tr>
<td>1,4</td>
<td>14.45</td>
<td>3,4</td>
<td>14.45</td>
</tr>
<tr>
<td>2,1</td>
<td>14.45</td>
<td>4,1</td>
<td>14.45</td>
</tr>
<tr>
<td>2,3</td>
<td>14.45</td>
<td>4,2</td>
<td>17.15</td>
</tr>
<tr>
<td>2,4</td>
<td>17.15</td>
<td>4,3</td>
<td>14.45</td>
</tr>
</tbody>
</table>

that $q_1$s decrease in a near-exponential fashion and the output traffic converges to the input traffic smoothly. Finally, the travel times are shown in Table 2.

3) Time-varying demands

Rush hour is characterized by time-varying demands and extrapolation errors might be significant. Simulation runs were initially conducted for small threshold values and 45 veh/min was determined to yield the "best" results. With initial queues zero, two sinusoidal demands of equal peak but different average values were considered — $130 + 16 \sin(0.05t)$ veh/min and $120 + 26 \sin(0.05t)$ veh/min. The resulting queues and flow rates from the former are shown in Figs. 47 — 48, and 49 — 50, respectively. Note that the queues increase during the rush hour and decrease at the end; and that the output traffic follows the input traffic closely. Finally, the travel times are shown in Table 3. Note that since they exceed the shortest delay-free travel times by approximately 50%, this demand is considered to have saturated the network. Further, during the rush hour, the external demands studied are in phase temporally. If they were in phase spatially, the congestion would be more serious and travel times would be longer.

The travel times for the second demand are shown in Table 4. Comparing Tables 3 and 4 one concludes that although the rush-hour peaks are the same,
Figure 47: Queues at nodes 1 to 4 under time-varying demands
Figure 48: Queues at nodes 5 to 8 under time-varying demands
All inputs = 130 + 16\sin(0.05t) (veh/min) 
- for input traffic
x for output traffic

Node 1

Node 2

Node 3

Node 4

Figure 49: Traffic at nodes 1 to 4 under time-varying demands
Figure 50: Traffic at nodes 5 to 8 under time-varying demands
Table 3: Travel times under heavy, time-varying demands—case 1

<table>
<thead>
<tr>
<th>S-D pair</th>
<th>Travel time</th>
<th>S-D pair</th>
<th>Travel time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2</td>
<td>20.15</td>
<td>3,1</td>
<td>20.35</td>
</tr>
<tr>
<td>1,3</td>
<td>20.35</td>
<td>3,2</td>
<td>20.15</td>
</tr>
<tr>
<td>1,4</td>
<td>20.15</td>
<td>3,4</td>
<td>20.15</td>
</tr>
<tr>
<td>2,1</td>
<td>20.15</td>
<td>4,1</td>
<td>20.16</td>
</tr>
<tr>
<td>2,3</td>
<td>20.15</td>
<td>4,2</td>
<td>20.35</td>
</tr>
<tr>
<td>2,4</td>
<td>20.35</td>
<td>4,3</td>
<td>20.16</td>
</tr>
</tbody>
</table>

Table 4: Travel times under heavy, time-varying demands—case 2

<table>
<thead>
<tr>
<th>S-D pair</th>
<th>Travel time</th>
<th>S-D pair</th>
<th>Travel time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2</td>
<td>18.38</td>
<td>3,1</td>
<td>18.50</td>
</tr>
<tr>
<td>1,3</td>
<td>18.52</td>
<td>3,2</td>
<td>18.34</td>
</tr>
<tr>
<td>1,4</td>
<td>18.38</td>
<td>3,4</td>
<td>18.34</td>
</tr>
<tr>
<td>2,1</td>
<td>18.38</td>
<td>4,1</td>
<td>18.34</td>
</tr>
<tr>
<td>2,3</td>
<td>18.38</td>
<td>4,2</td>
<td>18.50</td>
</tr>
<tr>
<td>2,4</td>
<td>18.53</td>
<td>4,3</td>
<td>18.34</td>
</tr>
</tbody>
</table>

the travel times, and hence the congestion are different — the higher the nominal demand, the longer the travel time.

4) Stochastic demands

The effect of the proposed control algorithm in handling stochastic demands is investigated. In this case, extrapolation errors in demand estimation might be significant, thus a small threshold value of 40 veh/min was employed since it yielded the "best" results among the threshold values tested.

In current traffic engineering practice, a typical demand is represented by a random process which is uniformly distributed around a constant value. Two demand-fluctuation situations are considered — ± 10%, which is commonly encountered [75], and ± 20%. Six average demand levels are evaluated in each case.
Table 5: Travel times under stochastic demands—Case 1

<table>
<thead>
<tr>
<th>S-D pair</th>
<th>External Demand (veh/min/node)</th>
<th>40</th>
<th>60</th>
<th>100</th>
<th>120</th>
<th>134</th>
<th>134.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2</td>
<td></td>
<td>12.32</td>
<td>14.48</td>
<td>14.81</td>
<td>15.14</td>
<td>15.73</td>
<td>19.62</td>
</tr>
<tr>
<td>1,3</td>
<td></td>
<td>14.33</td>
<td>14.6</td>
<td>14.97</td>
<td>15.22</td>
<td>16.07</td>
<td>47.35</td>
</tr>
<tr>
<td>1,4</td>
<td></td>
<td>12.31</td>
<td>14.45</td>
<td>14.78</td>
<td>15.1</td>
<td>15.8</td>
<td>19.33</td>
</tr>
<tr>
<td>2,1</td>
<td></td>
<td>12.32</td>
<td>14.48</td>
<td>14.81</td>
<td>14.91</td>
<td>15.57</td>
<td>18.98</td>
</tr>
<tr>
<td>2,2</td>
<td></td>
<td>12.31</td>
<td>14.46</td>
<td>14.78</td>
<td>14.86</td>
<td>15.95</td>
<td>18.15</td>
</tr>
<tr>
<td>2,3</td>
<td></td>
<td>14.32</td>
<td>14.54</td>
<td>14.9</td>
<td>14.86</td>
<td>16.56</td>
<td>41.66</td>
</tr>
<tr>
<td>2,4</td>
<td></td>
<td>14.33</td>
<td>14.58</td>
<td>14.96</td>
<td>15.3</td>
<td>16.4</td>
<td>46.04</td>
</tr>
<tr>
<td>3,1</td>
<td></td>
<td>12.31</td>
<td>14.46</td>
<td>14.77</td>
<td>15.17</td>
<td>16.32</td>
<td>18.45</td>
</tr>
<tr>
<td>3,2</td>
<td></td>
<td>12.30</td>
<td>14.43</td>
<td>14.74</td>
<td>15.2</td>
<td>16.42</td>
<td>18.26</td>
</tr>
<tr>
<td>3,3</td>
<td></td>
<td>12.31</td>
<td>14.45</td>
<td>14.78</td>
<td>14.94</td>
<td>16.02</td>
<td>18.98</td>
</tr>
<tr>
<td>4,1</td>
<td></td>
<td>14.32</td>
<td>14.54</td>
<td>14.89</td>
<td>14.92</td>
<td>16.71</td>
<td>41.71</td>
</tr>
<tr>
<td>4,2</td>
<td></td>
<td>12.30</td>
<td>14.43</td>
<td>14.74</td>
<td>14.9</td>
<td>16.34</td>
<td>18.04</td>
</tr>
</tbody>
</table>

and their relationship with mean travel time is shown in Table 5 and Table 6, respectively. Note that the times shown here are generally larger than those for the deterministic case (see Table 1) and increase more rapidly as demand increases. Also note that in the ± 10% case, substantial increases in travel times occur for the largest level considered, which is $r_D = 134.5$ veh/min; similar increases occur in the second case for $r_D = 130$ veh/min. The first case is further considered in detail in Fig. 51 — 52, which show substantial and continuous increases in some queue lengths, and in Fig. 53 — 54, which show various flow rates. Clearly, as the degree of randomness increases, the network capacity decreases — from 143 veh/min in the deterministic case to 130 veh/min in the second case considered here. Further, by comparing Tables 1, 5, and 6, it is noted that the average times increase correspondingly. Also note that travel times increase as the randomness in demand increases.
Figure 51: Queues at node 1 — 4 under stochastic, heavy demands
Figure 52: Queues at node 5 — 8 under stochastic, heavy demands
Figure 53: Traffic at node 1 — 4 under stochastic, heavy demands
Stochastic; all inputs = 134.5 (veh/min)

- for input traffic
- for output traffic

Node 5

Node 6

Node 7

Node 8

Figure 54: Traffic at node 5 — 8 under stochastic, heavy demands
5) Restricted Egress-Capacity Situation

The effect of limited exit capacity on the proposed control algorithm is investigated by employing a sinusoidal demand of $90 + 15 \sin (0.05t)$ veh/min at each source node and an output capacity of 100 veh/min at each destination node. Various threshold levels were tested for the “best” choice and 45 veh/min was selected.

The results are shown in Fig. 55 — 58. Note that the queues at all nodes increase during the rush hour and decrease at its end. As a result of excessive queues, the output traffic is greater than the input traffic at each node during the rush hour. Finally, the average travel times are shown in Table 7. Clearly, these are much larger than the corresponding delay-free times.

Generally speaking, congestion would form during the rush hour and decrease at its end. Further, with the reset mechanism, queues are distributed to “up-
Figure 55: Queues at node 1 — 4, limited egress-capacity case
All inputs = 90 + 15sin(0.05t)(veh/min)
Cexit = 100
- for q1
x for q2

Figure 56: Queues at node 5 — 8, limited egress-capacity case
Figure 57: Traffic at node 1 – 4, limited egress-capacity case
Figure 58: Traffic at node 5 — 8, limited egress-capacity case
Table 7: Travel times, with limited egress capacities

<table>
<thead>
<tr>
<th>S-D pair</th>
<th>Travel time</th>
<th>S-D pair</th>
<th>Travel time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2</td>
<td>35.58</td>
<td>3,1</td>
<td>80.31</td>
</tr>
<tr>
<td>1,3</td>
<td>80.26</td>
<td>3,2</td>
<td>37.16</td>
</tr>
<tr>
<td>1,4</td>
<td>36.56</td>
<td>3,4</td>
<td>38.13</td>
</tr>
<tr>
<td>2,1</td>
<td>35.56</td>
<td>4,1</td>
<td>36.49</td>
</tr>
<tr>
<td>2,3</td>
<td>37.18</td>
<td>4,2</td>
<td>81.53</td>
</tr>
<tr>
<td>2,4</td>
<td>81.54</td>
<td>4,3</td>
<td>38.13</td>
</tr>
</tbody>
</table>

stream” nodes and would not create serious congestion at downstream nodes. Finally, if the demand levels were larger and/or queues were not zero initially, the average travel times would become longer — ultimately approaching infinity for a sufficiently large demand.

4.3.2 Unbalanced-Demand Situation

Since the network is symmetric, it suffices to study an unbalanced traffic distribution at node 1 only. Here, two situations, one in which more traffic is heading for node 2 and one in which more traffic is heading for node 3 are considered. Since other situations can be considered as combinations of these two, a study of only these two situations should be sufficient. In each situation, two scenarios are investigated: a demand with a 10% imbalance and one with a 25% imbalance. Through experimentation, 45 veh/min and 40 veh/min are selected as threshold values for the former and the latter, respectively as they yielded the “best” results.

In the 10% demand-imbalance case of the first situation, six demand levels between 45 veh/min and 137.5 veh/min at each source node are studied, and the normalized (with respect to the external demand at each source node) demand distribution at node 1 is [0 0.4 0.3 0.3], where each column corresponds to a destination. The resulting travel times are shown in Table 8. Note that these times
Table 8: Travel times under unbalanced demands—Case 1

<table>
<thead>
<tr>
<th>S-D pair</th>
<th>External Demand (veh/min/node)</th>
<th>40</th>
<th>60</th>
<th>120</th>
<th>136</th>
<th>137</th>
<th>137.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2</td>
<td>12.0</td>
<td>13.99</td>
<td>14.0</td>
<td>15.37</td>
<td>18.19</td>
<td>21.78</td>
<td></td>
</tr>
<tr>
<td>1,3</td>
<td>14.0</td>
<td>14.02</td>
<td>14.0</td>
<td>17.82</td>
<td>40.43</td>
<td>59.13</td>
<td></td>
</tr>
<tr>
<td>1,4</td>
<td>12.0</td>
<td>13.99</td>
<td>14.0</td>
<td>14.81</td>
<td>18.0</td>
<td>21.77</td>
<td></td>
</tr>
<tr>
<td>2,1</td>
<td>12.0</td>
<td>14.0</td>
<td>14.0</td>
<td>14.74</td>
<td>15.98</td>
<td>18.12</td>
<td></td>
</tr>
<tr>
<td>2,3</td>
<td>12.0</td>
<td>14.0</td>
<td>14.0</td>
<td>14.02</td>
<td>15.15</td>
<td>17.23</td>
<td></td>
</tr>
<tr>
<td>2,4</td>
<td>14.0</td>
<td>14.02</td>
<td>14.0</td>
<td>19.83</td>
<td>37.2</td>
<td>54.57</td>
<td></td>
</tr>
<tr>
<td>3,1</td>
<td>14.0</td>
<td>14.02</td>
<td>14.0</td>
<td>17.66</td>
<td>39.83</td>
<td>60.98</td>
<td></td>
</tr>
<tr>
<td>3,2</td>
<td>12.0</td>
<td>14.0</td>
<td>14.0</td>
<td>14.06</td>
<td>15.79</td>
<td>17.79</td>
<td></td>
</tr>
<tr>
<td>3,4</td>
<td>12.0</td>
<td>14.0</td>
<td>14.0</td>
<td>14.04</td>
<td>16.34</td>
<td>19.01</td>
<td></td>
</tr>
<tr>
<td>4,1</td>
<td>12.0</td>
<td>14.0</td>
<td>14.0</td>
<td>14.75</td>
<td>17.21</td>
<td>20.49</td>
<td></td>
</tr>
<tr>
<td>4,2</td>
<td>14.0</td>
<td>14.02</td>
<td>14.0</td>
<td>19.84</td>
<td>37.04</td>
<td>54.04</td>
<td></td>
</tr>
<tr>
<td>4,3</td>
<td>12.0</td>
<td>14.0</td>
<td>14.0</td>
<td>14.03</td>
<td>16.2</td>
<td>19.55</td>
<td></td>
</tr>
</tbody>
</table>

increase substantially as the demand level approaches 137 veh/min. Obviously, this number must be the limit imposed on the network under this unbalanced-demand condition. In the 25% demand-imbalance case, the normalized demand distribution of [0 0.5 0.25 0.25] at node 1 is employed and the corresponding travel times are shown in Table 9. Observe that the increase in travel times become significant when the demand level approaches 124 veh/min, which can be considered as the limiting network capacity. Comparing Tables 8 and 9, it follows that travel times increase whereas network capacity decreases as the degree of demand-imbalance increases as would be expected.

In the second situation, normalized demand distributions of [0 0.3 0.4 0.3] and [0 0.25 0.5 0.25] are employed and the travel times are shown in Tables 10 and 11, respectively. Note from the former that travel times increase substantially if the demand level is greater than 139 veh/min, which should be the network capacity
Table 9: Travel times under unbalanced demands—Case 2

<table>
<thead>
<tr>
<th>S-D pair</th>
<th>External Demand (veh/min/node)</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>40</td>
<td>60</td>
<td>120</td>
<td>124</td>
<td>124.5</td>
<td>125</td>
</tr>
<tr>
<td>1,2</td>
<td>12.0</td>
<td>13.99</td>
<td>17.71</td>
<td>19.38</td>
<td>18.69</td>
<td>37.2</td>
</tr>
<tr>
<td>1,3</td>
<td>14.0</td>
<td>14.02</td>
<td>16.99</td>
<td>33.37</td>
<td>38.71</td>
<td>57.09</td>
</tr>
<tr>
<td>1,4</td>
<td>12.0</td>
<td>13.99</td>
<td>16.64</td>
<td>17.56</td>
<td>16.33</td>
<td>21.12</td>
</tr>
<tr>
<td>2,1</td>
<td>12.0</td>
<td>14.0</td>
<td>15.56</td>
<td>15.35</td>
<td>16.22</td>
<td>36.3</td>
</tr>
<tr>
<td>2,3</td>
<td>12.0</td>
<td>14.0</td>
<td>14.02</td>
<td>14.26</td>
<td>15.15</td>
<td>35.47</td>
</tr>
<tr>
<td>2,4</td>
<td>14.0</td>
<td>14.03</td>
<td>14.02</td>
<td>32.22</td>
<td>39.27</td>
<td>68.86</td>
</tr>
<tr>
<td>3,1</td>
<td>14.0</td>
<td>14.02</td>
<td>15.55</td>
<td>32.98</td>
<td>41.78</td>
<td>60.7</td>
</tr>
<tr>
<td>3,2</td>
<td>12.0</td>
<td>14.0</td>
<td>14.03</td>
<td>14.66</td>
<td>15.59</td>
<td>32.4</td>
</tr>
<tr>
<td>3,4</td>
<td>12.0</td>
<td>14.0</td>
<td>14.0</td>
<td>14.58</td>
<td>14.85</td>
<td>16.3</td>
</tr>
<tr>
<td>4,1</td>
<td>12.0</td>
<td>14.0</td>
<td>15.54</td>
<td>15.34</td>
<td>15.45</td>
<td>18.58</td>
</tr>
<tr>
<td>4,2</td>
<td>14.0</td>
<td>14.03</td>
<td>14.02</td>
<td>32.38</td>
<td>38.26</td>
<td>64.56</td>
</tr>
<tr>
<td>4,3</td>
<td>12.0</td>
<td>14.0</td>
<td>14.0</td>
<td>14.25</td>
<td>14.32</td>
<td>17.63</td>
</tr>
</tbody>
</table>

under this unbalanced-demand situation. Similarly from Table 11, it should be noted that the travel times become at least 50% longer than the corresponding delay-free times as the demand exceeds 129 veh/min. Clearly, this must be the network capacity under this demand condition. Comparing these two tables, it follows that travel times increase whereas network capacity decreases as the degree of demand-imbalance increases. Further, by comparing the results in both types of demand-imbalance situations, one concludes that the network capacity depends on the type of imbalance.

4.4 Conclusions and Discussion

A three-level multi-destination flow and routing control algorithm was proposed. Locally optimal flow rates into and out of a node were determined by the method of Chapter 2, and upstream and downstream traffic for each destination
Table 10: Travel times under unbalanced demands—Case 3

<table>
<thead>
<tr>
<th>S-D pair</th>
<th>External Demand (veh/min/node)</th>
<th>40</th>
<th>60</th>
<th>120</th>
<th>139</th>
<th>139.5</th>
<th>140</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2</td>
<td>12.0 13.99 14.0 14.07 15.13 67.81</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1,3</td>
<td>14.0 14.02 14.01 16.34 26.99 183.15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1,4</td>
<td>12.0 13.99 14.0 14.07 15.24 60.99</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,1</td>
<td>12.0 14.0 14.0 14.21 15.21 70.96</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,3</td>
<td>12.0 14.0 14.01 14.26 15.76 133.15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,4</td>
<td>14.0 14.02 14.0 14.67 23.85 150.55</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3,1</td>
<td>14.0 14.02 14.01 14.98 25.82 200.43</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3,2</td>
<td>12.0 14.0 14.0 14.07 14.94 146.16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3,4</td>
<td>12.0 14.0 14.0 14.07 15.04 136.66</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4,1</td>
<td>12.0 14.0 14.0 14.2 14.8 59.73</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4,2</td>
<td>14.0 14.02 14.0 14.67 23.38 150.32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4,3</td>
<td>12.0 14.0 14.01 14.27 15.17 120.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 11: Travel times under unbalanced demands—Case 4

<table>
<thead>
<tr>
<th>S-D pair</th>
<th>External Demand (veh/min/node)</th>
<th>40</th>
<th>60</th>
<th>120</th>
<th>129</th>
<th>129.5</th>
<th>130</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2</td>
<td>12.0 13.99 14.0 14.75 18.24 21.84</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1,3</td>
<td>14.0 14.02 16.35 22.62 32.94 57.32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1,4</td>
<td>12.0 13.99 14.0 14.31 17.84 21.83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,1</td>
<td>12.0 14.0 14.01 15.53 18.36 21.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,3</td>
<td>12.0 14.0 14.01 16.38 18.42 25.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,4</td>
<td>14.0 14.02 14.02 18.82 35.58 76.35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3,1</td>
<td>14.0 14.02 14.02 19.93 31.34 61.11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3,2</td>
<td>12.0 14.0 14.01 14.62 17.78 26.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3,4</td>
<td>12.0 14.0 14.0 14.18 17.38 26.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4,1</td>
<td>12.0 14.0 14.0 15.28 17.91 21.16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4,2</td>
<td>14.0 14.02 14.02 19.01 35.55 75.85</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4,3</td>
<td>12.0 14.0 14.01 16.33 18.0 25.31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
on each link were obtained through a proportional scheme and a LP formulation, respectively. Since the solution of the latter might not result in an even distribution of traffic on the downstream links with approximately equal costs, a procedure was employed to even out such traffic. Further, links were assigned as inadmissible with respect to some destinations to prevent vehicles from being routed back to their starting point. Finally, the concept of hop-count cost was introduced for inclusion when the external demand exceeded a threshold value and queues were empty to reduce traffic interaction and improve network performance.

Simulation studies for a highway network of a typical in-land city were conducted under deterministic-, stochastic-, and unbalanced-demand situations. In the first, both constant and time-varying demand cases as well as non-zero initial queues and limited egress-capacity situations were considered. In the stochastic- and unbalanced-demand situations, two scenarios were tested for each case and the effects of both randomness and imbalance in demands were explored.

The results in the deterministic, balanced-demand situation showed that the maximum demand level that the network could carry approached approximately 95% of the physical network capacity whereas those in the stochastic-demand situation revealed that this demand level was reduced — the greater the randomness the lower this level. In the time-varying situation, this demand level was also reduced since extrapolation errors were involved. The results in both limited egress-capacity and non-zero initial-queue situations indicated that the queues were "pushed" toward the upstream nodes during the rush hour and reduced in a near-exponential fashion at its end.

Two unbalanced-demand scenarios were investigated. The first involved more traffic heading for a destination to which only two routing paths were available and the second pertained to more traffic heading for a destination to which three
routing paths were available. The results showed that the general effect of unbalanced traffic was increased travel time under heavy demand levels. If the excessive demand was to the destination associated with more routing paths, the network capacity was larger since congestion could be reduced by distributing traffic onto more links. Further, it was found that the capacity was a function of the degree of imbalance in external demands — the higher the degree of imbalance, the lower the capacity.

Comparing the results obtained in the stochastic- and unbalanced-demand situations, it is found that the effect of randomness resembles that of imbalance in demands. In the stochastic case, demands might be unbalanced in level, whereas in unbalanced-demand case, demands are unbalanced in distribution. However, the effects in both cases are the same — larger extrapolation errors, increased travel times, more serious congestion, and reduced network capacity.

The results obtained in this simulation study showed that, under balanced-demand conditions, the network capacity could be high; however, under stochastic- and/or unbalanced-demand conditions, it is reduced as compared with the former situation. Further, congestion would be “distributed” to upstream nodes and vehicles at an entry ramp would be guided to other highways to avoid overcrowding the network. As a result, a journey through the network would be less congested. Finally, since the traffic was generally smooth, fuel consumption could be reduced.
CHAPTER V

SUMMARY, DISCUSSION, CONCLUSIONS, AND FUTURE RESEARCH

5.1 Summary

In general, a decentralized control algorithm is more reliable and computationally more efficient than a centralized one; however, time delays and slow convergence relative to the response characteristics of the network are generally present. The decision of whether to employ a centralized or decentralized control is usually dependent on the structure of the system to be controlled. In a transportation network, which covers a large geographic area, decentralized control would be preferred, as it is probably more reliable and more efficient and less costly than centralized control — very important factors in such a network. For these reasons, a coordinated, decentralized approach was employed in this study.

Asynchronous, rather than synchronous, control-interval updating was employed as this was easier to employ in a decentralized context, and a model involving two queues associated with each node was proposed. A corresponding objective function was formulated to achieve minimal congestion and smooth traffic flow and locally optimal solutions that could adapt to changing traffic conditions were derived for demand levels less than both the upstream and downstream capacities.

Two schemes were proposed if the demand exceeded either of these capacities. If it exceeded the upstream capacity, a goal that would both minimize $q_1$ and result
in smooth traffic flow was formulated, and a closed-loop solution was derived. If
the demand exceeded the downstream capacity, an objective function that would
both utilize the node storage and result in smooth traffic flow was proposed, and
a linear state-feedback solution was obtained.

A hypothetical queue, \( q_2 \), was introduced to allow a controller to store vehicles
at upstream nodes so that these nodes could collect them into a local queue, \( q_1 \), if
a threshold became excessive. In this way, congestion could be “pushed” toward
the entry point and the network would not become overcrowded.

A proportional scheme was employed for upstream traffic allocation and a LP
formulation was proposed for downstream routing. The goal associated with the
former was to ensure fairness among the upstream traffic on each link for each
destination, whereas that related to the latter was to minimize the downstream
travel time. A feature was included to even out the traffic on links with an app­
proximately equal cost to reduce the possibility of congestion. A linear model was
employed for the demands, queue sizes, and downstream costs, and a least-squares
algorithm with a variable forgetting factor was employed for the on-line estimation
and prediction of these quantities.

The control interval \( t_f \) associated with a downstream node was included in
the cost for upstream nodes. Normally, it was the time to clear queues; however,
if queues were zero at the beginning of a control interval, it was assigned a fixed
number (zero in light-demand conditions, non-zero in moderate- to heavy-demand
conditions), which could be considered as the cost of a “hop”. Basically, this cost
is included to prevent congestion from occurring rather than to route traffic after
it has occurred. It is one way of trading off between the travel time on a longer
link and the traffic interaction on a shorter link.

Simulation studies were conducted for both the single-destination and multi-
destination cases. The results from the former showed that in the constant-demand situation, the maximum flow that the network could handle approached the physical network capacity. In the time-varying demand situation, queues started to form at the beginning of a rush-hour and were reduced in a near-exponential fashion at its end — a realistic traffic phenomenon. Further, the network capacity depends on the type of the demand function — the higher the nominal demand, the lower the capacity.

In the multi-destination case, deterministic-, stochastic-, unbalanced-demand, and time-varying demand situations were studied. The results from the first showed that demands of up to 95% of the physical network capacity could be handled satisfactorily. If the demands exceeded this value, the network became congested and the mean travel times increased in an exponential manner. In the stochastic-demand case, balanced-demands (with a random component of ±10% of the nominal demand) of up to 90% of the capacity were acceptable. In contrast, Rule [33] showed that only 80% of the capacity was acceptable for this network when a centralized approach with fixed routes was employed. His stochastic-demand conditions were essentially the same as those employed here. In the unbalanced-demand situation, demands of up to 83% — 93% of capacity were acceptable, depending on the level and the type of imbalance. The results from the time-varying demand situation showed similarity between the multi-destination and the single-destination cases, i.e., exponential queue dynamics and lower capacity for higher nominal demand.

5.2 Discussion

The algorithm developed in this study was evaluated intensively for one important multi-input, multi-destination network. In this context, it appears essential
to discuss both factors which affected the results obtained and the extension of the concepts developed here to other networks.

Two factors which had a strong effect on network performance were the estimation errors and the prediction errors. In the deterministic, constant-demand case, the former converged to zero; however, in the time-varying and stochastic-demand situations, they did not converge and the network performance deteriorated accordingly. Such errors will always be present because of the stochastic nature of traffic; however, they could be reduced by employing historical data for prediction purpose.

Substantial link delays were incorporated into the network, and it was thus necessary to employ predictions of future arrivals at each node. This resulted in prediction errors because of the randomness and extrapolation involved. These errors, which could not be eliminated, resulted in substantial increases in delay in all cases. Historical data could also be employed to reduce such errors.

Normally, a problem involving a LP formulation is computationally intensive; in this study, \( l \times n \) variables were involved, where \( l \) and \( n \) were the number of downstream links and destinations, respectively. However, since a link was assigned as inadmissible with respect to some destinations, the number of variables was reduced. For the network considered in Chapter 4, the computation time at node 1 (where computations were one of the most intensive) was approximately 40 msec CPU time on a VAX 11/785. Further, since the solutions were in a simple closed form, few additional computations were necessary for the flow and routing control problem except in the determination of the next \( t_f \) where an iterative Newton method was employed. However, at worst, such a computation required 1 msec and was performed only once per control interval. Overall, the computations were "fast" with respect to the time scale of a transportation network, and should pose
no difficulties in a physical deployment.

In order to make the problem tractable, the queues, the link capacities, the upstream costs, and the downstream costs were aggregated. Consequently, the required information exchange between two neighboring nodes was reduced; however, any theoretical evaluation of network properties such as stability, or response times became difficult. A formulation that explicitly describes each queue, each link capacity, the upstream costs, and the downstream cost might be necessary to circumvent this difficulty; alternatively, a formulation could be employed in which each individual vehicle was continuously monitored and tracked. In either case, the problem would be exceedingly difficult for a network of any complexity.

A hop-cost was employed to account for traffic interactions and/or possible slow-downs due to merging conflicts and irregular destination distribution. Merging conflicts relate to situations in which two or more vehicles from different upstream links would occupy the same slot after merging if no speed-up/slow-down maneuvering was made, whereas irregular destination distribution corresponds to an uneven arrangement of vehicles heading for different destinations in a sequentially ordered traffic stream. Here the hop-cost and its threshold of application were selected empirically on the basis of results from simulation runs. The use of the cost thus obtained, resulted in improved network performance; however, it might be desirable to employ a mathematical approach to describe the dynamics of the merging conflicts and the irregular destination distribution. Normally, such dynamics are “random” and a stochastic model should be a natural choice. Further, if accurate measures of vehicle destinations were available at each node, accurate controls could be computed accordingly.

Both upstream and downstream costs were assumed to be constant over a control interval to simplify the derivation of the controls. In the simulation, how-
ever, these costs were time-varying over that interval; thus the controls employed were computed at each sampling instant so that the traffic could adapt to current conditions. Consequently, the solution became suboptimal and a stochastic model with time-varying parameters might be necessary to obtain a solution closer to an optimal one.

Node storage was assumed to be unlimited to simplify the derivation of the controls; in practice, such storage would be restricted and deadlock could occur. Then upstream vehicles would be blocked by those overflowing from downstream nodes and excessive congestion would result. This could be avoided by constantly monitoring the queue sizes for "overflow" and rejecting incoming traffic when necessary.

In a cyclic or multi-destination network, vehicles might be routed back and "looping" could occur. This corresponds to an "oscillation" in control terminology and an unstable condition. Thus, a link was assigned as inadmissible to some destinations so that a "loop" would not form. However, the network performance was deteriorated since the allocated input traffic for a specified destination sometimes exceeded the sum of the downstream capacities of all admissible links for that destination and congestion formed. As a result, the actual network capacity was reduced. Optimization procedures on a destination-by-destination basis were considered but not employed since it was difficult to obtain a closed-form solution. Further, the required computations would be intensive and sophisticated procedures might be necessary to determine the direction and the step size for the next iteration to ensure that the solution converged quickly while the numerical procedure remained stable. A unified solution method that does not have to artificially assign a link as inadmissible with respect to some destinations would be an interesting approach to evaluate.
The simulation study was conducted for a specific network under normal demand situations. For such a network, however, the demand pattern might be highly irregular, weather conditions might be adverse, and node-/link-failure might occur. The first can be characterized by a public event, such as a football game where vehicles would "swarm" to the same destination before such an event and "disperse" after it. Since the assignment of admissible destinations on a link was based on normal (i.e., "almost" balanced) demand situations, the network capacity would be reduced significantly in such an event. Such a problem may be alleviated by reconfiguring the network into an "event mode". This would involve reassigning the admissible destinations for each link at every node both before and after such an event.

Weather conditions such as fog and snow might affect the network performance by degrading some links. If this occurred, the capacity on the affected links would be reduced together with the network capacity. Such reductions can be lessened by adaptively reconfiguring the network; i.e., reassigning the admissible destinations for each link at every node.

Node-/link failure could occur in a network. Such a scenario could result from an accident and/or a natural disaster and vehicles would be "trapped" or "blocked" in the network. These situations could be handled by reconfiguring the network (e.g., the reassignment of a link as inadmissible to some destinations) and the rerouting of the vehicles already in the failed link and those approaching the link.

It is interesting to consider how the proposed algorithm could be employed for a general network. This would involve such factors as a network geometry quite different from that considered here, multiple lanes and multiple departure bays at each node. In regard to the first, since the proposed algorithm is decentralized,
it can be employed on a network with a general topology. In general, however, several studies must be conducted to determine the design parameters, such as $\Delta t_s$, the hop cost, and the associated threshold demand level. The resulting performance should generally be the same as those obtained in this study. However, the network capacity would be smaller for a network with large differences in distance among origin/destination pairs than the one with small differences since large extrapolation errors would be involved in the former.

The link capacities associated with a node might differ significantly and the accepted upstream traffic might exceed the sum of the capacities of the admissible downstream links; thus congestion might form. In such a case, it might be necessary to include the upstream demand, the downstream capacity, and the downstream cost in determining both $u^j_1$ and $u^j_1$ at each node, and to employ a destination-by-destination iterative procedure to obtain a solution.

A single-lane AHS was assumed in the development of the control algorithm. In some networks, two or more lanes might be involved in each direction and vehicles would be allowed to change lane. Normally, vehicles on the lane with a larger queue would “diffuse” into the lane with a smaller queue and, eventually, the queue sizes would tend to equalize. Therefore, the queues on both lanes can be “lumped” into one queue and the aggregated approach would still be applicable provided the “disturbance” due to the lateral maneuvering were accounted for.

Multiple-departure bays would probably not be involved in an AHS network due to land-use limitations. However, if so, a separate upstream lane for each downstream link would be required at each node and vehicles would move to the corresponding lane before they arrived at a node. Since a downstream link might be admissible only to some destinations, some queues on the upstream lanes might be much longer than others and travel times would be affected. A downstream
traffic-allocation scheme that includes these individual upstream queue sizes as well as downstream costs might be necessary.

5.3 Conclusions

A decentralized flow and routing control algorithm that is "best" for a general multi-destination network subject to all possible demand conditions is not yet within the state of the art. Here, a restricted case, a network in which every node is connected by the links with equal capacity was considered — the situation expected in practice. Although relatively high capacity was achieved with a deterministic, balanced-demand, it was reduced with practical time-varying, unbalanced-, and stochastic-demand conditions. Some of the difficulties encountered may be alleviated by use of historical data, especially under highly irregular demand conditions or when the differences in delay-free, link travel times are large. Additional refinements in the developed algorithm, such as the use of a stochastic model to predict merging conflicts, irregular destination distributions, and lateral maneuvering (when permitted) could result in reduced mean delay times.

Despite the difficulties encountered, the developed algorithm, which is relatively simple, easy to implement and not computationally intensive, resulted in an effective movement of traffic at high demand levels through a practical network. Further, the capacity achievable on a conventional freeway lane is approximately 33 veh/min under ideal conditions [76] whereas a flow of 45 veh/min (90% of the full physical capacity 50 veh/min) was achieved here with a stochastic demand (10% randomness) — a 36% improvement. If historical data were included in the estimation/prediction process, the improvement could be even greater.
5.4 Future Research

Several unsolved problems are specified here, and these might be profitable choices for future studies.

A least-squares estimator with a variable forgetting factor was employed in the estimation and prediction process. In this estimator, the forgetting factor \( f(n_s) \) was bounded within \((0,1]\); i.e., the error between the measured and the predicted values, \(e(n_s)\), and thus the maximum change in the estimated demand (if the estimator were a demand estimator) was bounded. If \( f_s \) were smaller, a larger bound could be employed; however, the average estimation error would become larger. In the current study, a small estimation error was considered most important and the variation of the predicted demand was bounded at a less-than-desired value. In a future study, approaches to overcome this limitation should be developed.

The optimal solutions were derived under the assumption that \( q_2 \) was unconstrained; however, it could be reset by the upstream nodes if it exceeded a threshold value. The effect of such a reset activity can be considered as a disturbance that occurs at random with a random magnitude. A stochastic formulation which includes these queue dynamics, a merging conflict model, an irregular destination distribution, a provision for lateral maneuvering, and time-varying parameters (e.g., for the upstream and downstream costs) could result in more efficient traffic control.

Several topics involving the application of the proposed algorithm to a network either with a general structure or under a variety of operating situations are possible candidates for future research. The former involves a network characterized by a variety of properties such as non-symmetrical network configuration, limited node storage, multiple departure-bay, multiple lane, and unequal link-capacity, as
were specified earlier in the discussion.

A network might operate under adverse conditions due to weather, controller reliability, and irregular demand patterns. The result would be reduced capacity and/or link blockage. Clearly, this would be a profitable area for future study. Another area would be the routing of high priority public vehicles (e.g., fire trucks or ambulances). One possibility would involve assigning a higher priority both to the link from which such vehicles arrive and to the link on which they proceed. In such a situation, it would probably be necessary to both slow down or stop other vehicles on some links and clear any existing congestion on others so that a path would be available for the quick movement of the emergency vehicles.

Finally, in view of the fast growth in the Artificial-Intelligence (AI) research area and its potential application to problems with incomplete information, recently developed theory may be profitably applied to achieve decentralized routing control [77] — [80]. A bridge that links AI with current control theory, especially in the large-scale control area, could be a profitable research domain for control theoreticians.
APPENDIX

FLOWCHARTS

The programs developed for the simulation studies include 7 modules [81]: NETWORK, NODEDYN, ALLOCATION, ESTIMATOR, REACT, SORT, and HPPLLOT. Here, NETWORK is the main program, NODEDYN determines both the aggregated controls and the queues for each node, ALLOCATION computes and allocates both the upstream and downstream traffic onto each link for each destination, ESTIMATOR estimates the input traffic, the queues, and the downstream costs, REACT reactivates (or resets) the estimator at the end of each control interval, SORT arranges the upstream link travel times in ascending order so that the input traffic on the link associated with the largest travel time is allocated first, and HPPLLOT plots the results. The hierarchy of these modules is shown in Fig. 59, and the flowcharts of NETWORK, NODEDYN, and ALLOCATION are shown in Fig. 60 — 62, respectively. The flowcharts for the remaining programs are not shown since ESTIMATOR is simply the implementation of equations (2.80) — (2.83), REACT only resets the clocks and the associated parameters \( \theta_1, \theta_0 \), the \( P \) matrix, and \( f(n_s) \), SORT employs the standard bubble-sort technique, and HPPLLOT involves device-dependent functions.

The same program modules were employed for both the single-destination and the multi-destination simulation studies. The differences between these studies were accounted for only in NETWORK by assigning different parameters for each
Figure 59: The hierarchy of the program modules

case. For a particular problem, only one set of these parameters was "active" and the other set was "masked" by comment statements.
Figure 60: The flowchart of NETWORK
Figure 61: The flowchart of NODEDYN
ALLOCATION

Enter

Normalize the weighting factors

Compute the upstream traffic for each dest.

Upstream traffic allocation?

Y

Determine upstream traffic on each link for each destination

N

Determine the downstream traffic for each destination

Setup constraint matrix, check for singularity (redundancy)

Compute the downstream traffic on each link for each destination

Even out the traffic on the links with approximately equal costs

Even out the traffic for the dest. of approximately equal costs

Adjust according to downstream congestion (multiply by w)

Return

Return

Figure 62: The flowchart of ALLOCATION
REFERENCES


