An Analysis of Personal Rapid Transit

by

Dylan Bradley Saloner

A dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy

in

Engineering-Civil and Environmental Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:
Professor Carlos Daganzo, Chair
Professor Samer Madanat
Professor Andrew Lim

Spring 2012
An Analysis of Personal Rapid Transit

Copyright 2012
by
Dylan Bradley Saloner
Abstract

An Analysis of Personal Rapid Transit

by

Dylan Bradley Saloner
Doctor of Philosophy in Engineering-Civil and Environmental Engineering
University of California, Berkeley
Professor Carlos Daganzo, Chair

Personal rapid transit (PRT) is a concept in which a fleet of automated vehicles operate on a network of grade-separated guideways, accessible on-demand at a fixed number of stations. The questions this research addresses include: How can merge-induced congestion be modeled and controlled? How do station configurations influence delay? How large must fleets be to accommodate a given demand profile? How should empty vehicles be dispatched to rebalance inventories at stations? While current literature about PRT tends to rely on simulation, this dissertation uses queueing theory as an analytical framework.
To my family.
## Contents

### List of Figures

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Elements of PRT</td>
<td>2</td>
</tr>
<tr>
<td>1.2 A Brief History of PRT</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Thesis Organization</td>
<td>5</td>
</tr>
<tr>
<td>2 Merges</td>
<td>6</td>
</tr>
<tr>
<td>2.1 Merges: Introduction</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Merges: Literature Review</td>
<td>8</td>
</tr>
<tr>
<td>2.3 A Queueing Model of Merges</td>
<td>9</td>
</tr>
<tr>
<td>2.4 Sizing Approaches</td>
<td>13</td>
</tr>
<tr>
<td>3 Stations</td>
<td>16</td>
</tr>
<tr>
<td>3.1 Stations: Introduction</td>
<td>17</td>
</tr>
<tr>
<td>3.2 Stations: Literature Review</td>
<td>20</td>
</tr>
<tr>
<td>3.3 Serial Station Model</td>
<td>20</td>
</tr>
<tr>
<td>3.4 Serial Station Analysis</td>
<td>23</td>
</tr>
<tr>
<td>3.4.1 Steady State Distribution</td>
<td>23</td>
</tr>
<tr>
<td>3.4.2 Serial Station Wave-Off Likelihood</td>
<td>25</td>
</tr>
<tr>
<td>3.4.3 Serial Station Delay</td>
<td>26</td>
</tr>
<tr>
<td>3.5 Serial and Parallel Operations Compared</td>
<td>26</td>
</tr>
<tr>
<td>3.5.1 Wave-off Likelihood</td>
<td>27</td>
</tr>
<tr>
<td>3.5.2 Delay for Alighting Passengers</td>
<td>27</td>
</tr>
<tr>
<td>4 Fleet Operations and Planning</td>
<td>29</td>
</tr>
<tr>
<td>4.1 Fleet Operations and Planning: Introduction</td>
<td>30</td>
</tr>
<tr>
<td>4.2 The Problem</td>
<td>31</td>
</tr>
<tr>
<td>4.2.1 Network</td>
<td>31</td>
</tr>
<tr>
<td>4.2.2 Demand</td>
<td>31</td>
</tr>
<tr>
<td>4.2.3 Network Configurations</td>
<td>32</td>
</tr>
<tr>
<td>4.2.3.1 A Simple Loop Configuration</td>
<td>32</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Station in Masdar, 2getthere PRT ........................................... 3
1.2 PRT Guideway at Heathrow Airport, Ultra PRT ......................... 4
1.3 Vehicles on a Test Track, Vectus PRT ....................................... 4

2.1 Physical Merge ........................................................................... 7
2.2 Point Process Representation of Merge ..................................... 10
2.3 Merge Decomposition ................................................................. 12
2.4 Merge Delay .............................................................................. 13

3.1 PRT Station Configurations ....................................................... 18
3.2 PRT Station Layout and Representation .................................... 21
3.3 Markov Chain Representation of a Serially Berthed Station With 4 Berths 22
3.4 Delay for Alighting Passengers: Serial vs Parallel ..................... 28

4.1 Simple Loop Configuration ........................................................ 32
4.2 Transition Diagram of a Birth Death Process ............................. 38
4.3 BCMP Model of an Uncontrolled Network .................................. 44
4.4 Isolated Station With Push Control Modeled as a Birth Death Chain 48
4.5 Stationary Distribution of Available Vehicles Under Push Control Model 51
4.6 Isolated Station With Pull Control Modeled as a Birth Death Chain 53
4.7 Fleet Size Required For Push and Pull ....................................... 55
Acknowledgments

A journey from its inception to its completion, the writing of this thesis was anything but automated. It would be a convenient abstraction to imagine that the author steps inside a private bubble, pushes a few buttons and with nothing but his thoughts to disturb him simply emerges at his destination three years later. Nothing could be less true.

In reality, this journey involved countless stops and starts, sweeping meanders, and the occasional breakdown. Fortunately, my friends and family have supported me and propelled me along the way. Without them this thesis would not have been possible. It is a pleasure to thank them here.

I am incredibly grateful to my advisor, Carlos Daganzo. He has indulged my pursuit of what can only be described as speculative research. I thank him for showing me the power of simplicity. More than that, I thank him for his approachability, his patience, and his good humor.

I am also indebted to Andrew Lim and Samer Madanat for graciously serving on my committee. I am especially grateful that they were able to home in on a glitch in my research at my qualifying exam. It has been rewarding to sort that out over the last two years. (Honestly.)

It has been a privilege to learn from the faculty of the Transportation Engineering program at U.C. Berkeley. Thanks also to the librarians at the ITS Library for staffing the best (transportation) library in the world. I cannot imagine a better place to have gone to graduate school.

The core of my learning experience has been my interactions with my fellow students. In particular I enjoyed the many spontaneous and wide-ranging conversations with my office mates Eric Gonzales, Yiguang Xuan, and Juan Argote.

I am grateful to Jorge Barrios for sharing his PRT simulator results and his thoughts on demand responsive transit.

I am thankful to my entire extended family and my in-laws for their sustained support. From my Granny’s sausage rolls to trips to New York, they have nourished me in every conceivable way.

Thanks to Mom and Dad for their unconditional love. They raised me to think independently and supported me in all my endeavors.

Thanks to my brothers, Brendan and Rowan, for lightening the mood, even (especially) when it was inappropriate to do so. Thanks also to Brendan for his feedback on the introduction chapter. Any remaining mistakes are, of course, his fault.

Finite permutations of letters seem completely incapable of conveying the infinite love and support I have received from my wife, Allison. Her encouragement and friendship have sustained me in the writing of this thesis.

Finally, I would like to thank my daughter, Cora, an inexhaustible source of joy and inspiration to me. I hope that by the time she turns sixteen, the idea of owning her own car will be absurd.
Chapter 1

Introduction
Personal rapid transit (PRT) is a nascent transit mode. Small automated vehicles run along grade-separated guideways, transporting passengers directly between stations in the network.

If not planned properly, delay can arise in at least three stages. First, vehicles may not be immediately available to a passenger at his station of origin. Second, congestion has the potential to impede vehicles as they enter or exit stations. Third, delay may occur in transit at merges along the guideway, where vehicle streams are interleaved.

This dissertation examines each component of delay and develops planning procedures to achieve acceptable performance. To contextualize the topic, this introduction provides a brief overview of PRT.

1.1 Elements of PRT

Dozens of PRT systems have been proposed over the last forty years. Although details of their design vary, they all share certain defining features.

Physically, PRT systems consist of:

- A fleet of small, automated vehicles. Vehicles are lightweight and electrically powered, generally with a carrying capacity between two and six passengers.

- A network of dedicated guideways. Designs call for grade-separated guideways to avoid interaction with pedestrians and automobiles at ground level. Generally, designs call for guideways to be elevated, although they may also be underground. Guideways tend to be significantly slimmer than those supporting conventional transit vehicles.

- Small, densely spaced stations. These stations are placed parallel to the main guideway letting through traffic bypass stations.

- An automated control system. The control system is responsible for dispatching and routing vehicle, as well as maintaining safe following distances. Users need only input their destinations.

The following features characterize PRT service:

- **On demand.** Service is typically available at all times of day. Empty vehicles wait at stations to be boarded, so typically passengers depart immediately without waiting for an available vehicle to arrive.

- **Direct to destination.** Passengers need not transfer vehicles to travel between any two stations on the network.

- **Nonstop.** The offline placement of stations enables vehicles to bypass intermediate stations, stopping only at their final destination.

- **Private.** Passengers travel either alone or with a small, private group.
1.2 A Brief History of PRT

Although the first true PRT system began operation in 2010, the concept can be traced back to 1964, when Donn Fichter first published a book called *Individualized Automated Transit and the City.*[13]

Three of the earliest prototypes were developed in the 1960s: the Aramis project (initiated in 1967 in France), the Computer-controlled Vehicle System (initiated in 1970 in Japan), and Cabintaxi (initiated in 1969 in Germany). Ultimately, Aramis and the Computer-controlled Vehicle System were cancelled because of concerns over the safety of control systems governing vehicle separation.[5, 24] Cabintaxi was discontinued for financial reasons.[5]

Charged with studying “new systems of urban transportation that will carry people and goods ... speedily, safely, without polluting the air, and in a manner that will contribute to sound city planning” the U.S. Department of Housing and Urban Development issued a report in 1968 proposing the development of PRT, among other alternatives.[26]

In 1973, the U.S. Urban Mass Transportation initiated the construction of a PRT system. This led to the construction of the so-called “Personal Rapid Transit” system in Morgantown, West Virginia. It has been in continuous operation since it opened in 1975. Despite its name, it is not a true PRT system. Although it is automated and provides demand-responsive routing, the system groups passengers into relatively large 20 passenger vehicles and does not guarantee direct to destination service.[15]

Recently, PRT has moved from concept to reality. As of May 2012, two PRT networks are in operation.

In 2010, a PRT system designed by the Netherlands vendor 2getthere began service in Masdar, a planned city in Abu Dhabi.[20] Figure 1.1 shows one of the two stations in operation.

![Figure 1.1: Station in Masdar, 2getthere PRT](image)

[1]
In 2011, Heathrow Airport in London began operations of its network. The system, designed by the British company Ultra, replaces a shuttle bus service connecting the parking lot to the airport terminal. Figure 1.2 shows the PRT guideway at Heathrow Airport threading through larger, pre-existing automobile infrastructure. The light weight of the vehicles allows the elevated guideways on which they run to be light and narrow. It would almost certainly not be possible to build conventional automobile or transit infrastructure in the same space.

![Figure 1.2: PRT Guideway at Heathrow Airport, Ultra PRT](image1)

The designs of 2getthere and Ultra both call for rubber-tired electric vehicles. However, other propulsion systems are possible. For example, the Korean vendor Vectus, which is developing a system under construction in Suncheon, South Korea, uses external linear-induction motors to propel vehicles, which are attached to the guideway. Figure 1.3 shows vehicles on a test track.

![Figure 1.3: Vehicles on a Test Track, Vectus PRT](image2)
Numerous other sites have been proposed for PRT networks and several other vendors have prototypes under development.

1.3 Thesis Organization

Planners of PRT systems must provision enough infrastructure in the form of vehicles and guideway so that the system performs acceptably. Allocating too few resources could result in unsatisfied passengers, but allocating more resources than required could affect the financial viability of a particular project. PRT system operators must develop control policies to make the most efficient use of the resources in the network. This thesis develops a set of analytical tools for the benefit of planners and operators of PRT systems.

We identify three factors that have the potential to undermine the performance of a PRT system. Each is assigned a chapter. Chapters are essentially modular, meaning that they can be read in any order and the symbolic notation resets.

Chapter 2 develops a method to assist planners in avoiding guideway congestion. We consider merges, locations where two guideways are joined, as the fundamental source of congestion. They have the potential to create excessive delay and generate vehicle queues that overflow allotted spatial buffers. An analytically driven approach based on a decomposition of queues produces methods to identify problematic merges. Performance standards limit both delay and the likelihood of buffer overflows.

Chapter 3 looks at station performance, where congestion can be a problem. In particular, we focus on the implications of vehicles blocking one another as a source of congestion. We compare the performance of two configurations, one in which vehicles access berths independently and one in which blocking is a serious concern. In the latter case, a Markovian queueing model forms the basis for a numerical procedure to size stations so delay is kept below a minimum service standard.

Chapter 4 looks at the problem of fleet control and planning. The problem at the heart of this chapter is how to size a PRT fleet so that it is unlikely that passengers encounter stations without available vehicles. The answer depends on the real-time control policy through which empty vehicles are redistributed between stations. We consider two simple, decentralized control policies, in which decisions to dispatch empty vehicles are based solely on inventory levels at individual stations. Queueing networks and Markovian models enable fast numerical optimization of the control. These policies are compared to benchmarks bounding fleet size above and below.

Chapter 5 concludes, summarizing the thesis and identifying topics for future research.
Chapter 2

Merges
2.1 Merges: Introduction

This chapter examines the process of vehicle merging. Physically, a merge is a location where two guideway branches meet to form one branch, depicted in Figure 2.1. From the perspective of vehicle dynamics, a merge is a process in which two vehicle input streams are woven together, the output of which is a single stream. It is the job of a merge controller to coordinate vehicles so that some safety standard is maintained as they pass through the merge point. Consequently, vehicles must sometimes be slowed (inducing delay) to accommodate traffic on the opposing input stream.

Merges can degrade system performance in several ways:

1. Merge delay adds directly to user cost.

2. Delay increases circulation time, reducing the productivity of the fleet.

3. Merges have the potential to cause queues on the incoming guideways to spill back and block other intersections in the network.

Thus, it is important to characterize certain features of the merging process such as the mean induced delay and the likelihood of buffer overflow.

Planners of PRT networks can use the analyses presented here as a tool to identify those merges with the highest potential to degrade system performance, adding merge capacity and queue storage to the network until the added infrastructure cost is balanced by the reduced cost of merge-related delay.

Alternatively, viewing a network as fixed, operators can use the merge analysis to explicitly model congestion and optimally price fares, thereby managing demand (and ensuring that too much congestion does not occur).

![Figure 2.1: Physical Merge](image)

*Vehicle move from left to right. Arrows depict vehicle ordering under first-come-first-merge.*

The remainder of this chapter is organized as follows. Section 2.2 provides an overview of relevant literature. Section 2.3 describes a novel approach to arrive at a previously discovered formula relating mean merge delay to a small set of parameters. Finally, Section 2.4 extends the analysis to ensure that queues on the approaches to a given merge rarely overflow their spatial buffers limits using a bound on the variance of the wait time.
2.2 Merges: Literature Review

In reality, merges are complicated processes, relying on complex communication and control systems. A precise description of merge dynamics could include such factors as inter-vehicle communication latency, sampling frequency in the control loop, and other hardware-specific features. Xu and Sengupta, for example, simulate the performance of a merge controller under communication constraints, with limits placed on maximum allowable jerk (the derivative of acceleration).[31]

Although simulations can assess performance under highly detailed scenarios, they do not yield the physical intuition that stems from parametric analysis. Furthermore, the time required to execute simulations can be a problem, as can the decision of when to terminate execution. This is particularly a problem for the investigation of low-likelihood yet high-impact events, such as buffer overflow events at merge approaches. Another shortcoming of merge simulations is its use when the analysis is to be embedded in larger network optimization problems. For instance, a network operator may wish to allocate flows across a network so that some performance metric (e.g. delay) is optimized. If given only a numerical table of merge performance, the output of simulation, optimization requires an iterative numerical procedure. If the number of merges in the network is large, the computational burden can be substantial.

By contrast, an explicit functional model of merges may allow an exact and fast solution to network optimization problems. It is therefore desirable to capture analytically the relevant characteristics of merges. Tanner’s model[30] (1962) was among the first to provide such a model of two input streams merging into a single one. Tanner considered inputs to the merges as outputs of $M/D/1$ queues with one input branch having absolute priority over the other. That is, vehicles in a “minor” stream must yield to those in a “major” stream, and may enter the major stream only when a large enough headway appears. This rule is intended in part to reduce ambiguity as to which vehicle has the right of way, thereby making the merges safer.

Although Tanner’s model may fairly describe freeway on-ramps, it would not apply as well to a PRT system. This is because the automated control of PRT enables equal prioritization (first-in-first-merge, or FIFM) of the input streams without safety concerns. FIFM produces lower average delay than absolute prioritization because the merge always accepts a vehicle when one from either of the two streams is waiting to merge. Thus, for a given cumulative arrival curve for both branches combined, the cumulative departure curve under FIFM will be earlier than that of an absolute priority merge at all points. Cowan[11] (1979) found an an expression for the mean delay incurred at equal priority merges with more general merge input streams, namely compound Poisson processes. (His intended application was not automated vehicles but rather the confluence of two single lane “rural” roads.) Section 2.3 presents a new and simpler derivation of Cowan’s solution under Borel distribution of inputs, but does not contain any novel results.

The literature review did not reveal any previous work on the variance of the merge wait time or sizing approaches to avoid overflow, the subject of Section 2.4.
2.3 A Queueing Model of Merges

Cowan used a Markovian framework to establish an equation relating merge delay, given a set of parameters characterizing the input streams of vehicles. The derivation in this section is based on an alternative visualization of the merge, which readily yields the result for Borel inputs with almost no algebraic manipulation.

Precision is sacrificed for simplicity by considering a merge model of three parameters: a headway parameter, $h$, defined as the smallest allowable temporal separation between any pair of vehicles, and the vehicle flows of the two upstream branches, $\lambda_1$ and $\lambda_2$.

It is assumed that the two input processes can be characterized as independent output processes of $M/D/1$ servers with service rate $\mu = 1/h$ and arrival rates of $\lambda_1$ and $\lambda_2$. The intuition here is that the randomness within the vehicle stream is the result of passenger arrivals, which can reasonably be assumed as Poisson and pairwise independent between any two stations. And so, when vehicles are dispatched from a given station, subject to some minimal safety headway, it is reasonable to model the resulting stream as the output of an $M/D(\mu=1/h)/1$ queue. Note, however, that only the wait times caused by the merge are relevant. Thus, the delay imposed by the (fictitious) upstream servers must be ignored. Later, it will be shown that given this assumption on input stream characterization, the discharge streams resulting from the merge operations are themselves outputs of $M/D(\mu=1/h)/1$ queues, and so the characterization of vehicle streams is consistent for “trees” of merges.

Figure 2.2 depicts a sample idealized merge process.
Boldface capital letters denote traffic streams, each described by a stochastic process. Separations between hollow dots describe time separations. Solid lines between streams $B$ and $D$, $C$ and $E$, and $F$ and $G$ indicate delays resulting from the deterministic servers and dotted lines indicate no delay. The filled black dots represent the input and output points of the merge. The gray rectangles represent the minimum safety headway buffers of duration $h$.

Although the input streams are two independent processes, for the purpose of analysis it will be helpful to instead think of them as derivative processes of a single Poisson process, depicted in Figure 2.2 as stream $A$. This process has flow equal to the sum of the flow into the merge, i.e. $\lambda_1 + \lambda_2$, and can generate two independent Poisson processes using the concept known as Poisson thinning as follows. For each point in $A$, a “coin flip” biased with probability $\lambda_1 / (\lambda_1 + \lambda_2)$ assigns it to process $B$; otherwise it assigns that point to process $C$. The well known result of Poisson thinning is that $B$ and $C$ are themselves independent Poisson processes, with parameters $\lambda_1$ and $\lambda_2$ respectively.

When processes $B$ and $C$ are fed into deterministic servers with service times of $h$, they yield the independent $M/D(\mu=1/h)/1$ output processes $D$ and $E$. Each point in $D$ and $E$, depicted as circles on a timeline, represents the time it would take a given vehicle to reach the merge point if unimpeded by the merge.

The idealized merge operation is simple. First, the incoming vehicle streams are ordered based on a first-come-first-merge priority, depicted in the figure by overlaying the
two processes into a single ordered stream, $F$. (This contrasts with the merging model developed by Tamer in which one stream has absolute priority over the other.) Next, $F$ serves as the input to a deterministic server with service rate $1/h$ representing the merge to ensure safe spacing as vehicles pass through the merge. The final result of the merge is $G$.

The ultimate goal of this section is to find an analytical expression for the mean merge delay, i.e., the delay incurred between inputs $D$ and $E$ and output $G$. (Because there is no delay in the overlay operation, this is equivalent to the mean delay between $F$ and $G$.) Define this as $\Delta(\lambda_1, \lambda_2, h)$, a function of the flow rates of the two upstream branches and the minimal allowable headway. It will be shown that

$$\Delta(\lambda_1, \lambda_2, h) = \delta \left( \frac{\lambda_1 + \lambda_2}{h} \right) - \frac{\lambda_1}{\lambda_1 + \lambda_2} \delta \left( \frac{1}{h} \right) - \frac{\lambda_2}{\lambda_1 + \lambda_2} \delta \left( \frac{1}{h} \right)$$  \tag{2.1}$$

where the function $\delta(\lambda, \mu)$ represents the well known formula for mean delay induced by an $M/D/1$ queue with arrival rate $\lambda$ and service rate $\mu$, given by:

$$\delta(\lambda, \mu) = \frac{\rho}{2\mu(1-\rho)}, \quad \rho = \frac{\lambda}{\mu}$$

As a step in verifying this equation, we first show that if $A$ is fed into a deterministic server directly, the output $H$ is the same as $G$, i.e. is unchanged by the addition of the thinning operations, the two ensuing deterministic servers and the overlay operation. The only difference between streams $G$ and $H$ is the “ordering” of points. (For example, note that in Figure 2.2, which has been drawn to scale, $G$ and $H$ are identical realizations of point processes, although the blue/green sequencing, indicating the input streams, is altered.)

This can be seen by applying induction. Consider a block of points in $H$, defined as a portion of the output process without a time gap between consecutive services, i.e. the set of customers served in a server’s busy period. More formally, let $t_0$ be a time point for a customer in $A$ that experiences no wait time in the deterministic server, so the map from $A$ to $H$ preserves that point and we write: $H(t_0) = t_0$. Any such customer is the first member of a block. Then, for some $m$, subsequent arrival time points $\{t_1 \ldots t_m\}$ in $A$ are said to belong to a block of length $m$ in $H$ if for all such points, the mapping from $A$ to $H$ satisfies $H(t_k) \leq t_0 + hk$ and $H(t_{m+1}) > (m+1)h$. We want to show that blocks are preserved under the merge operation, i.e. $G(t_k) \leq t_0 + hk$ for $0 \leq k \leq m$. By induction, it can be assumed that before $t_0$, that the blocks in $G$ and $H$ are identical. So, by the definition of blocks, the point $t_0$ will lag the last point of the preceding output block of $G$ by at least $h$ time units and therefore it will experience no delay in the merge process and thus $G(t_0) = t_0$. It is also clear that the deterministic server of the merge process between $F$ and $G$ will be continuously engaged for the subsequent inputs $\{t_1 \ldots t_m\}$, independent of the result of the Poisson thinning process and so $G(t_k) \leq t_0 + hk$ for all $1 \leq k \leq m$. It goes without saying that the first two deterministic servers will not shift the output
process of $G$ to the left and so $G(t_{m+1}) \geq H(t_{m+1}) > (m+1)h$. Therefore, this block is preserved under the merge process, and by induction, all blocks of $G$ and $H$ are identical.

As a consequence, the output of the merge is stochastically identical to the output of an $M/D(\mu=1/h)/1$ queue with arrival rate $\lambda_1 + \lambda_2$. (This is a fortunate result because the output may itself act as an input to a subsequent downstream merge. Thus, the result extends to a “tree” of merges. Note, however, that diverges do not preserve this property.)

Equation 2.1 can now be verified by decomposing the total delay as depicted in Figure 2.3.

Before proving Equation 2.1 we introduce further notation. Let $W_{XY}$ denote the random variable representing the wait time incurred in the transition from stream $X$ to stream $Y$ by a random point in process $X$. Also, let $1_B$ denote the Bernoulli indicator that takes a value of one if the Poisson thinning of stream $A$ assigns a given point to process $B$ and is zero otherwise.

Since $H$ and $G$ are identical, we begin by noting that $W_{AH} = W_{AG}$. Note that $W_{AG}$ can be decomposed as the delay at the merge, $W_{FG}$, plus the wait times between $B$ and $D$, if the thinning assigns the given point to stream $B$ (i.e., $W_{BD}1_B$) or the delay between $C$ and $E$ (i.e., $W_{CE} (1 - 1_B)$). In sum:

$$W_{AH} = W_{FG} + W_{BD}1_B + W_{CE} (1 - 1_B) \quad (2.2)$$

To find an expression for the mean merge delay, first isolate $W_{FG}$ and take the expectation of both sides:

$$\mathbb{E}(W_{FG}) = \mathbb{E}(W_{AH}) - \mathbb{E}(W_{BD}1_B) - \mathbb{E}(W_{CE} (1 - 1_B))$$

By definition, $\mathbb{E}(W_{FG}) := \Delta (\lambda_1, \lambda_2, h)$. Using the PASTA property of queues, we have that $\mathbb{E}(W_{BD}1_B) = \mathbb{E}(W_{BD}) \mathbb{E}(1_B)$, and likewise for the final expression. Recall that $1_B$ is a Bernoulli variable, and so $\mathbb{E}(1_B) = \frac{\lambda_1}{\lambda_1 + \lambda_2}$. The transitions from $A$ to $H$, from $B$ to...
and from $C$ to $E$ are all $M/D/1$ processes allowing us to substitute $\delta(\lambda_1 + \lambda_2)$, $\delta(\lambda_1)$, and $\delta(\lambda_2)$ for $E(W_{AH})$, $E(W_{BD})$, and $E(W_{CE})$, respectively. This yields Equation 2.1.

The region of convergence for $\Delta$ is given by the triangle bounded $\{\lambda_1, \lambda_2 : \lambda_1 > 0, \lambda_2 > 0, \lambda_1 + \lambda_2 < \frac{1}{h}\}$. Plots of mean delay versus $\rho_1$ (i.e., $\lambda_1$ with $h$ fixed at 1) for various values of $\rho_2$ with are depicted in Figure 2.4.

Figure 2.4: Merge Delay

### 2.4 Sizing Approaches

In addition to ensuring that the mean delay is low, another concern in planning merges is the possibility of queues backing up beyond acceptable spatial limits. For example, it is undesirable if a queue backs up past an upstream merge or diverge point, since such an event could cause the “gridlock” familiar in automobile networks. Real-time system-optimal control of the network might be able to reduce the occurrence of overflow events and thereby mitigate such effects. But such interventions would require shifting delay and vehicle storage upstream. As a conservative design standard, we propose that overflow events rarely occur, assuming no upstream control interventions. This section finds a closed-form bound on the minimum capacity of the approaches so that this design standard is satisfied.

Let us take stock of the parameters to be used in this section. First, it is assumed that each approach to the merge has a finite number of vehicles that can be stored before queues back up beyond prescribed limits, denoted as $n_1$ and $n_2$. Also assume we are given input flows of $\lambda_1$ and $\lambda_2$ along with a minimum safety headway, $h$, with the same characterizations of the input processes as described in Section 2.3. Finally, we impose
a service standard parametrized by \( \alpha \), the minimum acceptable likelihood that a vehicle from either approach overflows its buffer (e.g. \( \alpha = .001 \)). In this section, a heuristic test will be developed to determine whether the service standard is met as a simple closed form function of these parameters.

Instead of adhering to a strict first-come-first-merge policy, the merge controller may grant one of the streams priority if it is in danger of overflowing its buffer. Such a policy will not affect the mean merge delay: provided vehicles on both guideways are waiting to enter the merge, their sequencing will not alter the discharge stream. (As with Figure 2.2, the “coloring” of the output process might be altered, but the output point process will not.) This control policy enables vehicles to be stored in either buffer and so we can consider a single combined of size \( n_1 + n_2 \) to hold vehicles for both approaches as they wait to enter the merge.

If the distribution of the waiting time to merge, \( W_{FG} \), were known, the network planner could check to ensure that the \((1 - \alpha)^{th}\) percentile of the distribution is less than \((n_1 + n_2) h\). Unfortunately, exact calculation of the distribution is difficult. Instead, we seek a bound on the variance and then apply a Chebyshev inequality to bound the probability of overflow.

We calculate the variance of \( W_{FG} \), taking the variance of both sides of Equation 2.2:

\[
V(W_{FG}) = V(W_{AH}) + V(W_{BD1B}) + V(W_{CE}(1 - 1_B)) + 2[Cov(W_{BD1B}, W_{CE}(1 - 1_B))] - Cov(W_{AH}, W_{BD1B}) - Cov(W_{AH}, W_{CE}(1 - 1_B))
\]

(2.3)

The variances and covariances of Equation 2.3 can be decomposed further. Noting that the product of two independent random variables \( X \) and \( Y \) decomposes as \( V(XY) = V(X)E^2(Y) + V(Y)E^2(X) + V(X)V(Y) \), we have that

\[
V(W_{BD1B}) = V(W_{BD}) \frac{\lambda_1}{\lambda_1 + \lambda_2} + \frac{\lambda_1 \lambda_2}{(\lambda_1 + \lambda_2)^2} \left( \frac{\lambda h}{2 \mu (1 - \lambda h)} \right)^2
\]

(2.4)

The calculation of \( V(W_{CE}(1 - 1_B)) \) is similar.

Each of the terms \( V(W_{AH}), V(W_{AH}), \) and \( V(W_{AH}) \) represent the wait time variance of \( M/D/1 \) queues, the expression of which can be found analytically using the well known formula, \( V(X) = \phi_X''(0) - \phi_X'(0)^2 \), where \( \phi_X \) is the Laplace transform of a random variable \( X \).

For an \( M/D/1 \) queue with unit service rate and arrival rate \( \lambda = \rho \), the Laplace transform of the wait time is known to be [17]:
\[ \phi_{W(s)} = \frac{(1 - \rho) s}{s - \rho (1 - e^{-s})} \]

It follows then that the variance is:

\[ V(W) = \frac{\rho(2 + \rho)}{6(1 - \rho)^2} - \left[ \frac{\rho}{2(1 - \rho)} \right]^2 \]

We now have analytical expressions for all the terms in Equation 2.3 except for the covariance ones, which we now consider.

Decomposing the first covariance term:

\[ \text{Cov}(W_{BD1_B}, W_{CE} (1 - 1_B)) = \mathbb{E}(W_{BD1_B}W_{CE} (1 - 1_B)) - \mathbb{E}(W_{BD1_B}) \mathbb{E}(W_{CE} (1 - 1_B)) \]

The first term on the right side of the equation is zero because \( \mathbb{E}(1_B (1 - 1_B)) \) factors out by independence. And so we have that

\[ \text{Cov}(W_{BD1_B}, W_{CE} (1 - 1_B)) = -\left[ \frac{\lambda_1 \lambda_2}{(\lambda_1 + \lambda_2)^2} \right] \left[ \frac{\lambda_1 h^2}{2 (1 - \lambda_1 h)} \right] \left[ \frac{\lambda_2 h^2}{2 (1 - \lambda_2 h)} \right] \]

The only terms in Equation 2.3 that remain unaccounted for are the final two covariance terms. Unfortunately, they are difficult to evaluate precisely; however, they are clearly positive. Therefore we have a lower bound on \( V(W_{FG}) \). Although the bound is composed of the sum of several simple analytical expressions computed in this section, the expression is quite lengthy, and so it is not written out here in full. Instead, denote the bound as \( \tilde{V}(\lambda_1, \lambda_2, h) \) so that:

\[ V(W_{FG}) \geq \tilde{V}(\lambda_1, \lambda_2, h) \]

Applying Cantelli’s inequality (a one-sided extension of the Chebyshev inequality), we have a test for the capacity required on the approaches:

\[ P\{W_{FG}/h \geq (n_1 + n_2)\} \leq \frac{1}{1 + V(W_{FG})/ [h (n_1 + n_2) - \Delta (\lambda_1, \lambda_2, h)]^2} \]
\[ \leq \frac{1}{1 + \tilde{V}(\lambda_1, \lambda_2, h)/ [h (n_1 + n_2) - \Delta (\lambda_1, \lambda_2, h)]^2} < \alpha \]

If the last inequality does not hold, the merge capacity test fails and the network planner should modify the network until each merge passes the test.
Chapter 3

Stations
3.1 Stations: Introduction

PRT stations act as the interface between passengers and vehicles. Each station has a number of berths, which are spots where passengers can enter or exit vehicles.

Although a variety of berth configurations are possible, Figure 3.1 depicts two models that offer contrasting archetypes: (a) serial and (b) parallel configurations.

Idealizing their operations will allow us to develop numerically efficient procedures and facilitate performance comparisons. On one extreme, the serial configuration is typified by blocking. Under this setup, a vehicle can enter a berth only if preceding berths are empty, and following service completion, can exit the station only if subsequent berths are empty. On the other extreme is a parallel station in which blocking can be ignored. Here, it can be reasonably assumed that vehicles enter and exit the parallel berth station independent of the occupancy of other berths and traffic in the station.

This chapter examines the performance of serial and parallel station configurations under a range of arrival rates, loading times, and numbers of berths. The goal is to provide tools and insights for PRT planners to configure and size stations to perform acceptably under specified demand inputs.

Station design involves many complex tradeoffs. For instance, compared with the serial configuration, the parallel configuration reduces the potential for vehicles undergoing service (i.e. loading or unloading passengers) to block other vehicles entering or exiting the station. However, this improved performance comes at the cost of added space and infrastructure in the form of merges and diverges. These hardware costs depend significantly on the given vehicle-guideway interface. For example, the costs of parallel station hardware are expected to be higher for systems where the vehicle is captive to the guideway than for systems with rubber-tired vehicles. Explicitly modeling implementation-specific costs is beyond the scope of this thesis, but station designers can apply the techniques in this chapter to find the required station sizes for parallel and serial configurations. Then, inputting this into models of station cost would produce the minimum cost configuration with acceptable performance.
Figure 3.1: PRT Station Configurations
(a) serial configuration: vehicles in service block others waiting to enter and exit
(b) parallel configuration: vehicles enter and exit the station with nearly independent interactions

Arrows signify direction of flow. Input buffers to the left of the platform store incoming vehicles if the station is unable to immediately process them.

Before discussing what is meant by acceptable performance, let us briefly consider what happens at a PRT station and outline how we intend to model the behavior.

Two activities take place at stations: the unloading of arriving passengers from vehicles to the platform and the boarding of departing passengers from the platform into empty vehicles. Correspondingly, there are two broad concerns in the planning and operations of a station. First, unacceptable delay can result from queues of departing passengers if the number of berths for boarding is insufficient – even if there is an ample supply of empty vehicles. While this has the potential to be a problem, a higher priority concern (for reasons to be discussed soon) is the event that a station is unable to accommodate incoming vehicles to deboard passengers at their destination upon completion of a trip.

Ideally, we would be able to develop analytical methods to characterize station behavior without resorting to simulation. Yet, queuing models tend to be difficult to analyze and for serial stations, pure closed form expressions of performance are indeed elusive. The next best thing is analysis that can be solved with numerical methods, which yields faster and more accurate results than simulation, but the development of such procedures still requires some simplifying assumptions. Therefore the portion of this chapter based on numerical analysis (Sections 3 through 5) assumes minimal coordination of vehicles as they enter and exit the station. Furthermore, simplifying “worst case” assumptions regarding departing passengers are made to develop an analytical model for the delay
experienced by arriving vehicles. Although conservative, these results can be helpful to design systems where level of service is paramount.

The analytical models developed in this chapter will consist of four parameters (reducible to three by dimensional analysis). The flow of arriving vehicles and the mean dwell time at the berth are considered a priori constants (and can be reduced to a single parameter by considering their ratio, i.e. by choosing the mean dwell time as the unit of time). There are also two independent decision variables that can be varied in the planning of a station: the number of berths and the size of the input buffer. Given the a priori parameters, this chapter is concerned with investigating an analytical formulas and numerical methods to select decision variables so that service standards are met. In particular, two complimentary service standards will be examined, defined by their respective metrics: wave-off likelihood and delay.

If the input buffer to a station is full, arriving vehicles must be “waved-off”, forcing the vehicle to circulate in the network for a period of time, returning some time after a berth becomes available. This is problematic for at least three reasons. First, the time to return to the station may be substantial, adding directly to the delay of the passenger in the waved-off vehicle. (On a one-way loop network, for example, the unfortunate passenger would be forced to take a detour around the entire network!). Second, the circulation of waved-off vehicles will add to the flows across the network, increasing delays and the potential for buffer overflows at merge approaches, with attendant problems as discussed in the previous chapter. Third, the circulation time will reduce the effective number of available vehicles in the fleet, increasing the likelihood that at some stations the supply of empty vehicle for departing customers will be depleted, contributing to the delay of departing passengers. These are all serious concerns, and so it is prudent to impose a service standard dictating that the likelihood of a wave-off at any given station is very small (less than .0001, say).

Even if the wave-off likelihood is made very small, delay incurred by vehicles waiting for berths to become available can still be significant. Therefore, a separate service standard is imposed, limiting the sum of the mean delay experienced by vehicles waiting in the input buffer to access a berth plus the mean delay incurred by passengers in vehicles waiting to exit stations when blocked by vehicles in front.

The primary goal of this chapter is to develop computationally efficient models of station operations (i.e. without the aid of simulation) so that both parallel and serial configurations can be examined through the dual lenses of the aforementioned service standards.

The Markov model of serial stations to be developed in this chapter is used as a tool to provide insights regarding the capacity of a serial station under worst-case conditions. In particular, the capacity to serve vehicles is evaluated when the queue of departing passengers is inexhaustible, thus ensuring “fail safe” operation.

Understanding the performance of general configurations under control (how to direct vehicles in real time) is an important feature of station operations and planning, but it is too complex to be considered analytically. Rather, a simple “greedy” control strategy for
serial stations whereby arriving vehicles fill the first available berth is considered. More sophisticated near-optimal control strategies that hold back vehicles may yield lower delays, but it should be noted that the capacity of the simple control strategy will be achieved the same capacity as an optimal control. Since in reality better control policies can be designed (an underlying assumption of the higher-level model of immediate empty vehicle dispatching to be developed in Chapter 4 on fleet management), the design approach in this chapter is likely to preserve the flexibility to use a variety of control approaches.

3.2 Stations: Literature Review

Serially berthed PRT stations have been studied as early as Dais and York in 1977.\cite{12} The goal of that paper is simulation of the wave-off rate as a function of the number of berths in the station in conjunction with a simple third-order (i.e. jerk-constrained) vehicle following modeling. As with the model proposed in this section, passenger queuing effects of departures are ignored. Anderson \cite{6} simulates serial stations under more under a more complex vehicle following algorithm, and incorporates platform-side passenger behavior in the simulation.

The literature review did not reveal any studies of PRT station analyses that did not rely on simulation to provide results. However, analytical work has been undertaken for a similar problem with a slightly different application, specifically tandem queues at bus stops. (In this context, queueing of passengers is a less important feature of the system and so it is reasonable to ignore its effects.) Gu et al. \cite{18} parametrically analyze serialized bus stops with blocking and Erlangian service, soluble with up to three berths.

In this chapter a Markovian queueing framework will be used to model the operations of serially berthed stations. The transition matrix used to define this chain has a blocked structure similar to the class of chains described by Jain et al. \cite{21} with so-called phased bivariate state spaces. The reference describes numerical techniques to solve for the stationary distribution of such chains. The numerical procedure to be used to solve for the stationary distribution of the Markov chain to be developed is based on this approach (although does not follow it exactly).

3.3 Serial Station Model

In this section we turn our attention to forming a Markov model of the queueing process in an uncontrolled serially berthed station. Vehicles are assumed to stop just once at a single berth for a random interval of time while they unload and load passengers. The model to be analyzed ignores the effects of the departing passenger service times. That is, the time interval for which a vehicle is in service at a berth is independent of whether the vehicle is boarding a passenger or not.

The advantage of considering only vehicle movements is the reduction of the number of parameters in the model, making the stationary distribution soluble by conventional
established numerical procedures, such as those described by Jain et al. Although the model is incomplete in describing interactions with passenger queues, it is useful for investigating the performance of the station when demand is saturated. We can imagine an inexhaustible queue of passengers waiting on the platform side of a station. Each time a passenger alights from a vehicle, a departing passenger immediately boards. The model captures this premise by assuming that the time in which a vehicle occupies a berth is independent of the state of the departing passenger queue.

This conservative assumption allows for a bound on the capacity of the station. Thus, if the vehicle-side delay meets a particular service standard under such conditions, then performance will be improved when the passenger queue is undersaturated.

The performance implications of serial queueing to departing passengers is more difficult to model analytically and are left to simulation in the final section of this chapter.

The number of berths in a station is denoted by $n$. Vehicles are assumed to enter the station in an uncontrolled manner, “greedily” filling the furthest accessible berth.

To begin, we represent the state of the serial station as a pair of integers, $(a, b)$. The first element, $a$, denotes the position of the queue tail. The sign of $a$ reflects whether any berths are accessible: $a > 0$ signifies a queue of $a$ vehicles waiting to be served; by contrast, $a < 0$ indicates that $-a$ berths are currently accessible to incoming vehicles. The second element of the state, $b$, represents the number of vehicles currently undergoing service. For example, the empty state, when no vehicles are queued or unloading is represented as $(-n, 0)$. Figure 3.2 depicts two other examples of state representation at stations.

![Figure 3.2: PRT Station Layout and Representation.](image)

*Shaded vehicles represent those in service. Unshaded vehicles are finished unloading, but cannot relinquish their berth because they are blocked by a vehicle unloading in front of them.*
Aside from the empty state, \((-n, 0)\), the queue position is always larger than \(-n\) and the number of vehicles under service is always between 1 and \(n\), inclusive. Formally, we define \(\Omega\) as the set of all feasible states = \(\{(a, b) \mid -n < a, 1 \leq b \leq n, b-a \leq n\} \cup \{(-n, 0)\}\).

The condition that \(b - a \leq n\) reflects the physical constraint that the number of vehicles being served, \(b\), plus the number of accessible states, \(-a\), may not exceed the total number of station berths, \(n\).

Independent of previous states, the following rules govern transition from a particular state \((a, b) \in \Omega\) to a new state:

- An arrival occurs with rate \(\lambda\) and changes the state to

\[
(a', b') = \begin{cases} (a + 1, b) & \text{if } a \geq 0 \\ (a + 1, b + 1) & \text{if } a < 0 \end{cases}
\]

- A service completion occurs with rate \(b\mu\) and changes the state to

\[
(a', b') = \begin{cases} (a, b - 1) & \text{if } b \neq 1 \\ (\max(a, 0) - n, \text{ middle}(0, a, n)) & \text{if } b = 1 \end{cases}
\]

This two-dimensional state space can be coerced into a one-dimensional representation of non-negative integers under the bijective map

\[
(a, b) \mapsto \begin{cases} \frac{(a+a)(a+a-1)}{2} + b & \text{if } a < 0 \\ \frac{n(n-1)}{2} + an + b & \text{if } a \geq 0 \end{cases}
\]

Figure 3.3 illustrates the structure of the state transitions when \(n = 4\).

Figure 3.3: Markov Chain Representation of a Serially Berthed Station With 4 Berths

The green arrows represent service completions, which occur with rate variable rate, proportional to the number of vehicles in service. The blue arrows represent arrivals of vehicles to the station, which occur at a constant rate of \(\lambda\).
Before turning our attention to solving for the stationary distribution of this chain, we must make one final assumption to ensure that our problems is well posed. Denote $V_n$ as the maximum of $n$ independent exponential random variables with mean $1/\mu$. It is a well known fact that the expectation of $V_n$ is given by the formula

$$
E(V_n) = (1/\mu) \sum_{k=1}^{n} (1/k) \quad (3.1)
$$

(As an interesting aside, $E(V_n)$ is asymptotic to $(\log n + \gamma)/\mu$, where $\gamma = .577\ldots$ is Euler’s constant.) Therefore, the maximum capacity of any serial loading station is given by the number of vehicles that can be served in a full batch, $n$, divided, by the expected time to process a full batch, namely $n/E(V_n) = n\mu/\left[\sum_{k=1}^{n} (1/k)\right]$. Should the demand $\lambda$ exceed the capacity, the queue of vehicles to enter the station will grow without bound. Thus, it is will be assumed that $\lambda < n\mu/\left[\sum_{k=1}^{n} (1/k)\right]$.

### 3.4 Serial Station Analysis

In this section we analytically obtain the steady state distribution, $\{p_i\}$, which we will then use to compute the likelihood of buffer overflow and also use to compute the mean delay.

#### 3.4.1 Steady State Distribution

We find the steady state distribution using the concept of flow balance, which holds that in steady state the rate of transitions from a given state equals the rate of transitions into that state. By dimensional analysis no generality is lost in replacing $\mu$ with 1, and regarding $\rho$ as the ratio of $\frac{\lambda}{\mu}$. Then, the transition rate of the process depicted in Figure 3.3 is characterized by the following matrix (with subscripts indicating matrix dimensions for clarity):

$$
Q = \begin{bmatrix}
B_0 & C_0 & 0 & \ldots \\
B_1 & A_1 & A_0 & 0 & \ldots \\
B_2 & 0 & A_1 & A_0 & 0 & \ldots \\
B_3 & 0 & 0 & A_1 & A_0 & 0 & \ldots \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots \\
B_{n-1} & 0 & \ldots & 0 & A_1 & A_0 & 0 & \ldots \\
B_n & 0 & \ldots & 0 & A_1 & A_0 & 0 & \ldots \\
0 & A_{n+1} & 0 & \ldots & 0 & A_1 & A_0 & 0 & \ldots \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
& & & & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots
\end{bmatrix}_{\infty \times \infty}
$$

where:
\[ A_0 = \begin{bmatrix} -\rho & \cdots & -\rho \\ \vdots & \ddots & \vdots \\ -\rho & \cdots & -\rho \end{bmatrix}_{n \times n} \]
\[ A_1 = \begin{bmatrix} \rho + 1 \\ -2 & \rho + 2 \\ -3 & \rho + 3 \\ \vdots & \ddots \\ -n & \rho + n \end{bmatrix}_{n \times n} \]
\[ A_{n+1} = \begin{bmatrix} 0 & \cdots & 0 & -1 \\ 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix}_{n \times n} \]
\[ B_0 = \begin{bmatrix} \rho & -\rho \\ -1 & \rho + 1 & -\rho \\ -1 & \rho + 1 & -\rho \\ 0 & -2 & \rho + 2 & -\rho \\ -1 & \rho + 1 & \cdots & \cdots \\ 0 & \cdots & -2 & \rho + 2 \\ 0 & \cdots & \cdots & -3 & \rho + 3 \\ \vdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}_{m_n \times m_n} \]
\[ B_{l}(i, j) = \begin{cases} -1 & i = 1, j = l^2 - l \\ 0 & \text{otherwise} \end{cases}, \quad (n \times m_n) \]
\[ C_0 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & -\rho & 0 & \cdots & 0 \\ 0 & 0 & -\rho & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & -\rho \end{bmatrix}_{m_n \times m_n} \]

where \( m_n = \frac{n^2 - n + 2}{2} \)
(To clarify, the number of nonzero elements in $C_0$ is $n-1$.)

Let $p$ be the row vector representing the steady state distribution and partition it corresponding to the blocking of $P$ so that $p = (p_0, p_1, p_2, \ldots)$, where the size of $p_0$ is $m_n$ and that of $p_i, i \geq 1$ is $n$. For $k > 1$, we claim that their exists an $n \times n$ matrix $R$ such that $p_k = p_1R^{k-1}$.

Note that $A_{n+1}$ can be expressed as $u_1^Ta_2$ for row vectors $u_1 = [-1, 0, \ldots, 0]$ and $u_2 = [0, \ldots, 0, 1]$.

We may assume that for $k > 1$ $p_k = p_1R^{k-1}$ for some $R$. Such an $R$ must satisfy the following:

$$A_0 + A_1R + A_{n+1}R^{n+1} = 0_{n \times n}$$

To find $R$, we can use an iterative procedure:

$$R_{\text{new}} = -A_1^{-1}(A_0 + A_{n+1}R^{n+1})$$

The procedure terminates when $\sum_{i,j}\left| (R_{\text{new}})_{ij} - (R_{\text{old}})_{ij} \right|$ falls below some small convergence threshold. Given the approximation of $R$, we can now solve for $p_0$ and $p_1$ using two boundary equations and a normalization of the probability mass. Denoting $x$-dimensional row vectors of zeros and column vectors of ones as $0_x$ and $1_x$ respectively we have the following system of equations:

$$p_0B_0 + p_1\sum_{k=1}^{n} R^{k-1}B_k = 0_{m_n}$$

$$p_0C_0 + p_1A_1 + p_1R^nA_{n+1} = 0_n$$

$$p_01_{m_n} + p_1\sum_{k=1}^{\infty} R^{k-1}1_n = p_01_{m_n} + p_1(I - R)^{-1}1_n = 1$$

(3.2)

The stationary distribution, encapsulated by $p_0$ and $p_1$, can be found now using basic linear algebra, with the first two equations above representing a homogeneous linear system of dimension $m_n + n$ and the third equation isolating the solution space as a single vector.

### 3.4.2 Serial Station Wave-Off Likelihood

With the stationary distribution in hand, we can formulate the buffer overflow service standard.

The model developed thus far describes a system with infinite queue capacity, which at first glance may not seem to model finite buffer overflow. However, provided that the likelihood of buffer overflow is very small (an imposition of the service standard) an infinite state model is appropriate.
The service standard requires that the probability that a buffer of size \( d \) is full, denoted as \( q_d \) must be less than some small value, \( \epsilon \):

\[
q_d := 1 - \left[ p_0 1_{m_n} + p_1 \sum_{k=1}^{d} R^{k-1} 1_n \right] \leq \epsilon
\]

To find the required buffer size, given \( n \) and \( \rho \), it is computationally inexpensive to iteratively increase \( d \) until \( q_d \leq \epsilon \). The results of this procedure for various values of \( n \) and \( \rho \) are shown in Section 3.5.

### 3.4.3 Serial Station Delay

Delay is the sum of the wait time a vehicle experiences to enter a berth plus the wait to exit the station once service is complete. If a vehicle arrives at state \((a, b)\) and finds no accessible berths, its expected delay will be the sum of:

- mean service time of vehicles currently in service, \( \sum_{k=1}^{b} \frac{1}{k} \)
- mean service time of each of the \( \left\lfloor \frac{a}{n} \right\rfloor \) batches ahead of the arriving vehicle, \( \left\lfloor \frac{a}{n} \right\rfloor \sum_{k=1}^{n} \frac{1}{k} \)
- mean service time of all \( a \mod n \) vehicles ahead of the arriving vehicle in its own batch, \( \sum_{k=1}^{a \mod n} \frac{1}{k} \) (or 0 if \( a \mod n = 0 \)).

Summing over the stationary probabilities of all feasible states the mean stationary delay is expressed as:

\[
W = \sum_{(a,b) \in \Omega} p_{(a,b)} \times \begin{cases} 
\left( \sum_{k=1}^{b} \frac{1}{k} + \sum_{k=1}^{a \mod n} \frac{1}{k} + \left\lfloor \frac{a}{n} \right\rfloor \sum_{k=1}^{n} \frac{1}{k} \right) & a > 0 \\
\left( \sum_{k=1}^{b} \frac{1}{k} \right) & a \leq 0
\end{cases} \quad (3.3)
\]

Calculating the delay can be now be done by summing the terms in the state space using the sequencing shown in Figure 3.3 until numerical convergence is achieved. Alternatively, using a Jordan normal decomposition of \( R \), it is possible to find a closed form expression for \(3.3\).

### 3.5 Serial and Parallel Operations Compared

This section describes an alternative configuration: parallel berthing at queues. Parallel berthing requires an alternative hardware configuration, whereby a vehicle can enter or exit each berth independent of the states of other berths. Moreover, the idealized parallel station ignores traffic conflicts that might occur entering and exiting berths.
3.5.1 Wave-off Likelihood

Note also that exiting vehicles are not blocked at parallel stations and thus, it is reasonable to model the parallel station as a simple $M/M/n$ queue. To model an idealized parallel configuration with $n$ berths, we can apply the well known $M/M/n$ model. As noted in Gross[17], the stationary distribution for this model is:

$$
\tilde{p}_j = \begin{cases} \frac{\rho^j \tilde{p}_0}{j!} & (0 \leq j < n) \\ \frac{\rho^j}{n^{j-n}\tilde{p}_0} & (j \geq n) \end{cases}
$$

where $\tilde{p}_0 = \left( \frac{r^n}{n!(1-\rho)} + \sum_{j=0}^{n-1} \frac{r^j}{j!} \right)^{-1}$, and $r = \rho n$

For given $n$ and $\rho$, it is straightforward to find the minimum $d$ such that the service standard is satisfied by incrementing $d$ until the following inequality is satisfied:

$$
\tilde{q}_d = 1 - \sum_{k=0}^{d} \tilde{p}_k < \epsilon
$$

3.5.2 Delay for Alighting Passengers

Having mandated that the wave-off likelihood is low, the input buffer considerations of a parallel station can be ignored when looking at delay and an $M/M/n$ model is appropriate. The well known formula for queue wait time in such a system is given as:

$$
W_q = \frac{r^n}{n!(n\mu)(1-\rho)^2} \tilde{p}_0
$$

This is plotted in Figure 3.4 against the serial delay, as calculated in Section 3.4.3.

Obviously, for fixed $n$ and $\rho$, a parallel configuration is preferred to the corresponding serial one. However, for non-operational reasons, serial stations might be preferred should they provide lower spatial requirements at platforms and stations or if they cost less to build, especially when switching hardware is expensive and vehicles are held captive to the guideway.

Hence, the preference for parallel or serial will depend on the costs of building each and the expected demand, requiring complex tradeoffs that may not be so easily modeled. Furthermore, when evaluating performance from a purely operational perspective, the merits of parallel over serial hardware can be quite sensitive to $\rho$, and so the accuracy of this parameter is very important.
Figure 3.4: Delay for Alighting Passengers: Serial vs Parallel

Delay of serial and parallel station configurations are plotted as a function of the offered load for. Each color represents a different number of berths, from 1 to 5. For fixed \( n \) parallel configurations dominate their serial counterpart, which blow up much sooner.

The findings from this chapter can be summarized as follows. The operational costs of serially configured stations are much higher than parallel stations and there should be a compelling motive to use the former over the latter. If serial configurations must be used, judicious management of platform side passengers can improve performance significantly.
Chapter 4

Fleet Operations and Planning
4.1 Fleet Operations and Planning: Introduction

Having considered the microscopic features of PRT networks, merges and stations in Chapters 2 and 3, we are now in a position to look at PRT from a more macroscopic perspective. The question at the core of this chapter is *How many vehicles are needed in a particular network to guarantee satisfactory performance?*

To answer this, we consider the intertwined problems of operating and planning a fleet of vehicles at the network level. The concern of *operations* is redistributing empty vehicles between stations in real-time so that for a given network and fleet size, a performance metric (e.g., mean passenger delay or likelihood of a passenger arriving to a station without available vehicles) is minimized. For a particular operating scheme, *planning* involves finding the minimum number of vehicles so that a service standard is achieved.

Because planning and operations are related problems, it is sensible to consider them holistically. Ideally, we would be able to devise an optimal controller and then plan accordingly. (Here, optimal implies that no other controller could improve the performance with the same number of vehicles.) However, fully optimizing empty vehicle control is thought to be an intractable problem. Instead a simple class of decentralized control schemes is considered. By *decentralized*, it is meant that decisions to dispatch empty vehicles are based solely on inventory levels at individual stations, rather than the state of the entire network. Specifically, we explore two schemes dubbed *push* and *pull control*. The basic idea is that if the number of vehicles at a station reaches certain upper or lower thresholds, vehicles are pushed (forwarded) to or pulled (requested) from nearby stations.

Despite the simplicity of the approach, it will be seen that these controllers perform reasonably well. The primary advantage of focusing on decentralized operations is that the planning and operations problems can be combined and optimized over a restricted parameter space. As with previous chapters, an analytical approach is taken. This yields quick, accurate results without resorting to simulation techniques that pervade the literature.

This chapter is organized as follows. Section 4.3 reviews the literature dealing with empty vehicle management in PRT networks. Section 4.2 provides a more detailed statement of the problem, and introduces notation and modeling assumptions. Section 4.4 describes the theory underlying stochastic modeling techniques to be employed in modeling the dynamics of network control. Section 4.5 develops quick and simple approximate methods to calculate upper and lower bounds on system performance, i.e. the fleet size required for particular networks. Section 4.6 contains the key analytical results. It describes push control and models it using two techniques. First, the control method is modeled precisely over a complete stochastic network. However, a second less complicated model yields nearly identical results, motivating a decentralized approach to modeling (and controlling) the network. With this in mind, Section 4.7 applies a similar decentralized model to pull control. Finally, the push and pull models are compared. It is found that pull control requires a significantly smaller fleet than push control.
4.2 The Problem

This section formalizes the fleet management problem, discussing modeling assumptions and introducing notation.

4.2.1 Network

A PRT network consists of \( n \) stations connected to one another with segments of one-way guideway.

Although between a particular pair of stations numerous route choices may be feasible, the travel time between station \( i \) and station \( j \) is encapsulated as a single value, \( t_{ij} \). Over all station pairs, \( t \) takes the form of an \( n \times n \) matrix. (This is a modest abuse of notation since \( t \) needs to store just \( n^2 - n \) parameters, since for all \( i \), \( t_{ii} \) is zero.)

No two snowflakes nor realized travel times between a pair of stations are identical, but modeling the latter with a single parameter is reasonable. Ignoring the effects of congestion along the guideway, \( t_{ij} \) simply represents the minimum travel time among all routes between station \( i \) and station \( j \). Such an assumption is certainly justifiable for early proposed PRT installations in which few vehicles operate at low demands. Alternatively, if significant congestion along the guideway is anticipated, \( t_{ij} \) should be interpreted as the mean travel time from station \( i \) to station \( j \) under the system-optimal routing. System-optimal is defined as the routing that minimize mean travel time averaged over all passengers in the network such that demand is met. (Note that the system optimal solution under stationary routing probabilities could be modeled explicitly using the formula for merge delay developed in Chapter 2. This is fairly straightforward, but beyond the scope of this thesis.)

Furthermore, although Chapter 2 considered the stochastic effects of interaction between vehicles along the guideway, travel times in this chapter are deterministically modeled since the variance from congestion is assumed to be much less than the total travel time for a particular trip. Similarly, the delay entering and exiting stations is ignored. This is justified because applying the standard for station operations from Chapter 4 will result in near zero delay. Even if significant delay is anticipated, for trips terminating at station \( j \), the mean delay incurred entering station \( j \) could be added to \( t_{ij} \) for all \( i \). Likewise, for trips originating at station \( i \), the delay incurred exiting station \( i \) could be added to \( t_{ij} \) for all \( j \).

4.2.2 Demand

Demand over a network is characterized by a sequence of requests for immediate departure by a single individual (or group small enough to fit in a single vehicle) over a particular origin-destination station pair. The possibility of ride-sharing between more than one group or individual is ignored.
Requests from station $i$ to station $j$ comprise a Poisson process with rate $\lambda_{ij}$. Although the demand rate over a real-world network will certainly vary with time, our analysis considers stationary demand corresponding to peak demand. This reflects planning for the worst-case, when the number vehicles required to satisfy demand over a sustained time period would be highest. As with travel times, demand is encapsulated in an $n \times n$ matrix, $\lambda$, which effectively holds $n^2 - n$ parameters since the diagonals are zero. It is also assumed that the demand processes are independent across stations.

For notational convenience, define $\lambda_i := \sum_j \lambda_{ij}$ and $\lambda_i := \sum_j \lambda_{ji}$ as the trip rates from and into station $i$ respectively. For further convenience, denote $\Lambda$ as the total demand across the network, namely $\Lambda := \sum \sum \lambda_{ij}$.

4.2.3 Network Configurations

It will be helpful to apply the methods to be developed on a canonical network. Perhaps the simplest PRT network of interest is a one-way loop configuration, which we briefly present in this subsection. Simple loops will also prove to be a useful foundation in considering general PRT networks.

4.2.3.1 A Simple Loop Configuration

A simple loop configuration is depicted in Figure 4.1. Vehicles travel around the network in a single direction.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure4_1.png}
\caption{Simple Loop Configuration}
\end{figure}

For further convenience, we use the $\oplus$ to signify advancing and $\ominus$ to signify reversing through the station index in the tour loop so that the result is taken modulus $n$. So for example we have

\[ i \oplus 1 := \begin{cases} 
  i + 1 & \text{if } i < n \\
  1 & \text{if } i = n 
\end{cases} \]

and

\[ i \ominus 1 := \begin{cases} 
  i - 1 & \text{if } i > 1 \\
  n & \text{if } i = 1 
\end{cases} \]
The advantage of examining a simple loop configuration is twofold.

First, it provides a network specification with few parameters. In particular, a homogeneous loop can be described using three parameters: the number of stations \( n \), total demand \( \Lambda \), and loop travel time \( \tau \). The demand is \( \lambda_{ij} = \Lambda / (n - 1)^2 \) for \( i \neq j \) and the travel time is \( t_{ij} = \tau (j \ominus i) / n \). Furthermore, applying dimensional analysis the travel time \( \tau \) parameter can be eliminated, thereby expressing the entire network with just two parameters. This facilitates comparisons to other control schemes or modes. For example, a bus can be optimized analytically over this small parameter space enabling comparison to PRT.

Second, the decision space for redistribution of empty vehicles is much more restricted under a loop configuration. This makes reasonable the heuristic that each empty vehicle movement is between a pair of adjacent stations. Provided each station has at least one empty vehicle, an empty vehicle movement between station \( i \) and \( j \neq i + 1 \) has the same effect as \( j \ominus i \) simultaneous empty vehicle movements (one between \( i \) and \( i + 1 \), one between \( i + 1 \) and \( i + 2 \) \ldots and one between \( j - 1 \) and \( j \)). The primary difference is that the many-movement tactic achieves the same outcome as a single movement in \( 1/(j - i) \) time. Thus, it is reasonable to consider only empty vehicle movements between adjacent stations for single loop systems.

4.2.3.2 General Networks

We would like to consider general networks while preserving some of the simplifying properties and heuristics of a one-way loop network. To this end, a portion of the analysis for general networks contrives a one-way loop as defined by the traveling salesman problem (TSP). The TSP is a well studied problem inspired by the routing problem a traveling salesman might face when visiting a list of cities. Given distances between each pair of cities, the task is to find the shortest route that visits each city exactly once and returns to the origin city. In the case of a PRT network, the solution to the “cities” are station nodes and instead of distance we are interested in the path minimizes total time around the loop.[1]

For notational convenience, the station indexes are re-ordered so that \( \langle 1, \ldots, n \rangle \) is a solution to the TSP.

By introducing the TSP loop, in some sense general networks can be reduced to a one-way loop configuration. This notion will be expanded in future sections.

---

[1] It should be noted that although this chapter is concerned with developing computationally inexpensive methods, finding the exact solution to the traveling salesman problem over a large network is notoriously difficult (NP hard, in fact). This would thus seem to undercut the efficiency of methods to be developed, which are based in part on the TSP solution. However, most real world PRT networks proposed for the near term consist of a small number of one-way sub-loops, each of which is a de facto sub-solution to the TSP. (That is, the order of the stations along a particular sub-loop can be assumed to be preserved under the TSP solution.) For such networks, the TSP can be solved (or at least approximated) with low computational effort. A more detailed discussion of the TSP and associated heuristics is beyond the scope of this thesis.
4.2.4 Goal

Over the long run, the number of departures from a station will not balance with the number of arrivals. Even if the long-term arrival and departure rates happen to be equal, the inventory of available vehicles at a particular station will fluctuate. Without intervention the variance of the vehicle stock will grow in proportion to elapsed time. Therefore, empty vehicles (without passengers) must be moved between stations to balance inventory levels and guard against the possibility that stations run out of vehicles available for arriving passengers.

This chapter is concerned with devising a control policy to move empty vehicles. Individual trip requests are made in real-time, meaning that the empty vehicle controller has no knowledge of future trip requests aside from long term demand rates as given by $\lambda$. The controller takes as input the current state of the network: the inventory levels at each station and trips, empty or full, currently in progress. Trip information can be encapsulated by the destination of each trip and the time remaining until the trip is completed. The output of the controller is the dispatch of an empty vehicle from one station to another.

Devising a controller is embedded in the broader problem of sizing the fleet so that a performance standard is met. Here, performance of the controller is measured by the stationary likelihood that an arriving passenger finds a vehicle available at her station of origin.

More formally, if the system is run continually, the proportion of time during which station $i$ has $k$ vehicles available will converge to some value $0 \leq \pi_k(i) \leq 1$. Collectively, these values form the stationary distribution of the network. For a particular network and control policy, we seek the smallest $f$ so that $\pi_0(i) \leq \tilde{\pi}_0$, where $\tilde{\pi}_0$ is a standard of our choosing. (We set $\tilde{\pi}_0 = .01$.)

Note that mean delay could be another appropriate choice of performance metric, but the methods employed to model the network-level dynamics lend themselves more directly to the likelihood metric. In any case, this distinction is of little consequence since both metrics are reasonable and a sufficiently low likelihood of unavailable vehicles implies will result in low delay.

To recapitulate, the parameter space of this fleet planning problem on a network of $n$ stations is defined by two $n \times n$ matrices with $\lambda$ denoting inter-station demand and $t$ denoting inter-station. The ultimate goal of this chapter is to look at how fleet size $f$ varies under various control policies and demand parameters.

4.3 PRT Fleet Management: Literature Review

The literature on fleet management in PRT systems is scant. The large majority of papers dealing with empty vehicle control strategies rely on simulation.

Andréasson considers a three-stage heuristic to route empty vehicles.\cite{Andreasson} The first stage involves some initiation procedures; the second stage allocates vehicles to waiting passen-
gers based on longest wait times; the third stage reallocates empty vehicles based on minimum running distance using a heuristic to approximate the transportation problem. Performance is evaluated with a simulation.

Lees-Miller develops a fluid limit bound on the number of vehicles required to meet capacity, which can be solved with linear programming. The work develops a proactive vehicle redistribution approach based on sampling and voting. This method moves empty vehicles based on a nearest neighbor rule for known requests. It employs a rolling horizon approach to simulate future requests and proactively moves vehicles accordingly. This approach compares favorably to that of Bell and Wong, described below. Performance is evaluated with simulation.

In general, simulations are opaque, requiring substantial time to run, particularly when looking over multiple dimensions of the parameter space. Attempts to optimize control based on simulation are especially problematic because the decision space explodes with fleet size, the number of stations in the network, and the length of the time horizon under consideration. As a result, attempts to optimize control using simulation tend to be computationally challenging, and vulnerable to statistical inaccuracies and local optima traps. These problems are compounded further when considering the broader problem of fleet size planning.

Bell and Wong develop a rolling time horizon algorithm to dispatch taxis, although it could be applied to a PRT network. Under this setup, each vehicle, empty or full, has a list of stations it is assigned to visit. The control dictates that an arriving passenger is assigned the vehicle that minimizes the pickup time. The pickup time for a given vehicle and passenger request is defined as the time until the vehicle reaches the destination of its final assignment plus the time to move from its final destination to the passenger’s origin. Assignments are never revised. A vehicle remains idle at the destination of its final assignment until the next time it is assigned to a customer.

The Bell and Wong strategy is good at limiting empty vehicle movement, but fails a very simple thought experiment. Imagine a two station PRT network with positive demand from station 1 to station 2 but zero demand in the return direction. Here, the optimal control is obvious: as soon as a vehicle drops off a passenger at station 2, it should immediately return to station 1. Yet the Bell and Wong algorithm is reactive, requiring a passenger arrival to reposition vehicles. If there are no passengers waiting at station 1, vehicles will accumulate at station 2. The push and pull control methods to be developed will be optimal in this simple example.

4.4 Methods

This section overviews methodologies to be used as the basis for modeling the stochastic behavior of empty vehicle movement. Subsequent sections will connect the theory described in this section to the empty vehicle management problem.
4.4.1 Queueing Networks

Queueing networks will be proposed as a framework for modeling vehicle movement in PRT networks. This section provides an overview of some important theoretical results.

4.4.1.1 Gordon-Newell Networks

A network of \( m \) queues is called a Gordon-Newell network (or a closed Jackson network) if it meets the following conditions:

1. The network is closed. (Customers neither enter nor leave the network.)
2. All service times are exponentially distributed.
3. A customer completing service at queue \( i \) moves to a new queue \( j \) with probability \( P_{ij} \) such that \( \sum_{i=1}^{m} P_{ij} = 1 \)
4. The utilization of all the queues is less than one.

The Gordon-Newell Theorem \[16\] gives the equilibrium distribution of states in a Gordon-Newell network of \( m \) queues, each with exponential service rate \( \mu_i \), and with a total population of \( K \) individuals with state space \( S(K,m) = \{ k \in \mathbb{Z}^m \text{ such that } \sum_{i=1}^{m} k_i = K \text{ and } k_i \geq 0 \forall i \} \) where \( k_i \) represents the length of queue \( i \). Then the stationary probability distribution is given by

\[
\pi(k_1, \ldots, k_m) = \frac{1}{G(K)} \prod_{i=1}^{m} \left( \frac{e_i}{\mu_i} \right)^{k_i}
\]

where \( e_i \) is the visit ratio, found by solving the simultaneous equations

\[
e_i = \sum_{j=1}^{m} e_j p_{ji} \quad \text{for each } 1 \leq i \leq m
\]

and \( G(K) \) is the normalizing constant given by

\[
G(K) = \sum_{k \in S(K,m)} \prod_{i=1}^{m} \left( \frac{e_i}{\mu_i} \right)^{k_i}
\]

4.4.1.2 BCMP Networks

Baskett, Chandy, Muntz, and Palacios extended Gordon-Newell networks with the eponymous BCMP Theorem.\[8\] BCMP networks admit three additional types of queues as nodes in addition to single exponential servers (SER):

1. Infinite Server (IS). Here, an infinite number of servers is available.
2. Processor Sharing (PS). Here, each waiting job receives an equal share of processing capacity.

3. Last Come First Serve Preemptive Resume (LCFSPR). Here, jobs are received on a last come, first serve basis and work done on preempted jobs is not lost.

The service time distributions at queues of the last three types is quite general, restricted only to the class of Coxian distributions. However, the mean service times are the only inputs to the BCMP equations. Additionally, the BCMP theorem allows multiple customer types and service rates dependent on the number of customers in the queue. The BCMP Theorem [8] holds that the equilibrium distribution of the entire queueing network is the product of the respective distributions of each node, up to a constant $G$. For a network of $m$ nodes and $K = \sum_{i=1}^{m} k_m$ passengers in circulation, the distribution is written as:

$$\pi(k_1, \ldots, k_m) = \frac{1}{G(K)} \prod_{i=1}^{m} \pi_i(k_i)$$

The form of $\pi_i$ corresponds to the type of queue at node $i$. This thesis will employ networks that use a combination of SER and IS servers only.

Although its theoretical lineage traces back to transportation problems, which were the focus of Newell’s research [27], the BCMP Theorem has been applied most often to the analysis of computer and communications networks.[22] While there do appear to be applications outside of communication networks (for example, see [?] as an overview of BCMP applied to operations management) this thesis is believed to be the first application of BCMP networks to PRT systems or shared fleet transportation systems, more generally.

### 4.4.1.3 Numerical Procedures

The simple form of the joint distribution of a BCMP network is betrayed by the constant $G$, which cannot itself be written in closed form (except for the case of Gordon-Newell networks) and generally requires a numerical procedure to find. There are a number of methods to determine $G$, including Gibbs sampling and a variety of convolution procedures. The method used in this thesis is a mean-value-analysis algorithm that circumvents the need to find $G$ directly. The specific procedure employed is based on the work of Reiser and Lavenberg[29]. They generalized Buzen’s Algorithm[10], developed for Gordon-Newell networks, to BCMP networks.

In essence, Reiser and Lavenberg’s iterative algorithm is based on two principles. First, in a closed network with $K$ customers an arriving customer observes a queue length that is statistically equal to the queue length with $K - 1$ customers. Second, Little’s formula can be applied throughout the network. The details are substantial and are not presented here.
4.4.2 Birth Death Processes

While queueing networks provide a powerful way to holistically model a class of stochastic networks, a so-called birth-death process can, in some circumstances, approximate the most important features of networks using a more convenient formulation. This section describes the theory of birth-death processes, while the approximation heuristics of networks will be described later in the context of PRT networks in Section 4.6.

A birth-death process is an example of a continuous time, discrete state Markov process. By discrete it is meant that the family of random variables indexed by time \( \{X(t) : 0 \leq t < \infty\} \) take integral values. We will be concerned with processes where the transition probability function is stationary, i.e.

\[
P_{ij}(u) := P\{X(u + v) = j \mid X(v) = i\}, \ i, j \in \mathbb{N}
\]

is independent of \( v \geq 0 \).

The birth-death process is so named because transitions either increase or decrease the state by one (resulting in a “birth” or a “death” respectively). Specifically, the transitions rates are specified so that

\[
P_{ij}(u) = \begin{cases} 
\ell_i u + o(u) & \text{as } u \to 0 \text{ when } j = i + 1, i \geq 0 \\
\mu_i u + o(u) & \text{as } u \to 0 \text{ when } j = i - 1, i \geq 1 \\
0 & \text{else, } j \neq 1
\end{cases}
\]

Essentially, the birth-death process is like a random walk, except transitions take place in continuous rather than discrete time. The transition diagram for a general birth death process is depicted in Figure 4.2.

![Transition Diagram of a Birth Death Process](image)

**Figure 4.2: Transition Diagram of a Birth Death Process**

To find the stationary distribution of this process denoted as \( \pi_k \), apply cuts between states \( i \) and \( i + 1 \) and apply the global equations so that the probability flows across the cuts are balanced. This yields

\[
\ell_k \pi_k = \mu_{k+1} \pi_{k+1}
\]

Using recursion, the state probabilities can be expressed in term of the initial state \( \pi_0 \), which itself can be determined through normalization, so that
\[ \pi_k = \pi_0 \prod_{i=1}^{k-1} \frac{\ell_i}{\mu_{i+1}} \]
\[ \pi_0 = \left(1 + \sum_{k=1}^{\infty} \prod_{i=1}^{k-1} \frac{\ell_i}{\mu_{i+1}}\right)^{-1} \]  

\[ (4.1) \]

4.5 Fleet Size Benchmarks

This section provides heuristics to bound and approximate the required fleet size. First, we will cast the problem in the limit of deterministic fluid flow. This will give us the fleet size to serve demand at capacity, as well as provide a base for methods to be developed later. Then, introducing stochastic elements, we will use best case and worst case assumptions on empty vehicle access to provide further upper and lower bounds on fleet size. These bounds will be useful as bases of comparison for models to be developed later in this chapter.

4.5.1 Lower Bound: Fleet Size Under Deterministic Demand

We can find a lower bound to the fleet planning problem by sizing the fleet just large enough to accommodate sustained deterministic demand. Specifically we seek a set of empty vehicles routing flows that minimize fleet size with bounded delay. This subsection reproduces the work of Lees-Miller.

Instead of thinking of passenger demand and vehicle dispatch processes as discrete and random, it is helpful to view them in this context as continuous and deterministic.

The flow of vehicles can be thought of as fluids being pumped through two parallel sets of pipes, one set for full vehicles and the other set for empty vehicles. Each station takes as input one pipe of empty vehicles and one pipe of full vehicles from every other station in the network for a total of \(2(n-1)\) inflow pipes per station. Similarly, each station outputs \(2(n-1)\) pipes. At each station the total fluid flow passing between input and output pipes must be conserved since vehicles are neither created nor destroyed. (There are no “reservoirs” at stations because demand and supply are instantly matched.)

The flow along the full vehicle pipes is governed by the demand matrix \(\lambda\), which is given. Our job then is to find flow rates through the set of empty vehicle pipes so that flow at each station is conserved and the total amount of fluid in the system is minimized. The empty vehicle fluid flow is denoted by a nonnegative \(n \times n\) matrix of empty vehicle flows, \(\epsilon\). That is, \(\epsilon_{ij}\) represents the number of empty vehicles per unit time that flow from station \(i\) to station \(j\). The number of vehicles in the system, denoted as \(f\), is the product of total flow and travel time, summed over each pair of stations. Accounting for the flow conservation and the non-negativity of empty vehicle flow, the optimization problem can be expressed as
\begin{equation}
\min_{\epsilon_{ij}} \quad \sum_{i \neq j} \left( \epsilon_{ij} + \lambda_{ij} \right) t_{ij} \\
\text{s.t.} \quad \sum_{i \neq j} \left( \epsilon_{ij} + \lambda_{ij} \right) - \sum_{i \neq j} \left( \epsilon_{ji} + \lambda_{ji} \right) = 0, \forall i \\
\epsilon_{ij} \geq 0 \quad \forall i, j
\end{equation}

Optimization problems of this type can be solved as an LP. First, it is convenient to re-write the equations using vector notation. Let \( t, \lambda, \) and \( \epsilon \) denote vectors that flatten the matrices \( t, \lambda, \) and \( \epsilon \) by ordering the elements in a single list. Let \( \sigma \) denote the vector of surplus vehicle demand at each station so that

\[ \sigma_i := \sum_{j \neq i} \left( \lambda_{ji} - \lambda_{ij} \right) \]

Then Equation (4.2) can be re-written as

\begin{equation}
\min \quad t' \epsilon \\
\text{s.t.} \quad A \epsilon = \sigma \\
\epsilon \geq 0
\end{equation}

This particular linear program is a \textit{minimum cost network flow problem} (MCNF) and can be solved efficiently using the network simplex method. Denote \( \hat{\epsilon} \) as the solution to the MCNF of (4.3) If the fleet size is less than \( \hat{f} := \sum_{i \neq j} \left( \hat{\epsilon}_{ij} + \lambda_{ij} \right) t_{ij} \) the system demand will exceed capacity causing vehicle shortages and unmet demand. It should be noted that for demand-balanced configurations, \( \hat{\epsilon}_{ij} = 0, \forall i, j \), and so \( \hat{f} = \sum_{i \neq j} \lambda_{ij} t_{ij} \). For the case of a homogeneous loop with uniform demand, \( \hat{f} = \Lambda \left( \frac{n-1}{2n} \right) \).

### 4.5.2 Lower Bound: Empty Vehicle “Teleportation”

Another approach to lower bounding the fleet size admits stochastic demand but indulges the fantasy that empty vehicles can be “teleported” between stations without delay. (Even in this fantasy, teleportation is an experimental technology and doing so with passengers inside vehicles is considered too dangerous.) In other words, there is a pool of empty vehicles instantaneously accessible from each station. Needless to say, this will perform better than even an optimal empty vehicle controller and thus provides a lower bound on vehicle performance.

Given a fleet size of \( f \), the system functions as an \( M/D/f \) queue. Customers in the queue, i.e. passengers, arrive as a Poisson process with demand rate equal to the total demand in the network \( \Lambda \). Service, is then provided by one of \( f \) vehicles servers. Service time is the realization of a single one-way trip. The service rate is therefore the inverse
of the mean trip time, weighted by demand, i.e. \( \mu^{-1} = \bar{t} = \frac{1}{n} \sum (\lambda_{ij} t_{ij}) \). The number of customers in the system is the number of full vehicles in use.

The problem of appropriately sizing the number of servers in a standard queueing system is well studied. Using a simple, well-known expression [17] for approximating the number of servers as a function of demand and service rate in queues produces a lower bound approximating the number of servers required:

\[
f_{\text{LB}} \approx \Lambda \bar{t} + \beta \sqrt{\Lambda \bar{t}} \quad \text{(4.4)}
\]

Expanding the expression from Equation 4.4, the lower bound is written as

\[
f_{\text{LB}} \approx \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_{ij} t_{ij} + \beta \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_{ij} t_{ij}}
\]

Here \( \beta \) is a constant related to the standard of service, i.e. the probability of nonzero delay, which we require to be less than \( \tilde{\pi}_0 \). Then, by a well-known result of Newell[28] \( \beta \) is related to \( \tilde{\pi}_0 \) as

\[
\tilde{\pi}_0 = \frac{\phi (\beta)}{\phi (\beta) + \beta \Phi (\beta)}
\]

where \( \phi (\cdot) \) and \( \Phi (\cdot) \) are the PDF and CDF of a standard normal random variable. Kolesar and Green note that this can be approximated by setting \( \beta \) to the \((1 - \tilde{\pi}_0)^{1/2} \) quantile of the standard normal distribution [23].

### 4.5.3 Upper Bound: Fleet Decentralization

Next we form an upper bound on the fleet size. In contrast to the lower-bounding assumption of a centralized fleet, here we assume that the fleet must operate in a decentralized manner. Specifically, each station has its own fleet of vehicles, which it cannot share with other stations.

Again we can make use of the standard approximation from queueing theory to determine the appropriate number of servers for each “sub-fleet”. For each sub-fleet \( i \), the customer demand is the rate of trips originating from station \( i \), namely \( \lambda_i \). Upon arriving to its destination, a vehicle in subfleet \( i \) has no choice but to immediately return empty to station \( i \). The mean time a server (vehicle) is in use is therefore the time a vehicle takes to reach its destination plus the time it takes to return. This average, weighted by demand, is equal to \( \frac{1}{\lambda_i} \sum_{j=1}^{n} \lambda_{ij} (t_{ij} + t_{ji}) \). The \( \lambda_i \) terms cancel out and summing over the sub-fleets, the fleet size required to meet the service standard is upper bounded as

\[
f_{\text{UB}} \approx \sum_{i=1}^{n} \left\{ \sum_{j=1}^{n} \lambda_{ij} (t_{ij} + t_{ji}) + \beta \left( \sum_{j=1}^{n} \lambda_{ij} (t_{ij} + t_{ji}) \right) \right\}
\]
4.6 Push Control

As noted in the literature review, the analysis of empty vehicle control considered elsewhere either relies on simulation or performs poorly under some simple counterexamples. This section describes a simple, decentralized control strategy that does not suffer such limitations.

Under what we dub push control, station $i$ accepts empty vehicles until its inventory reaches a threshold $b_i$ which is considered a decision variable (or a function thereof). Once saturation has been reached with $b_i$ vehicles, additional empty vehicles are automatically redistributed ("pushed") to station $i \oplus 1$, which is defined as the station that follows station $i$ in the TSP tour. If downstream stations are themselves saturated, a single push may trigger a cascade of pushes. The basic idea of this scheme is that empty vehicles pushed off from stations with excess inventory will eventually find their way to those stations with a dearth of empty vehicles.

If a passenger encounters a station with no empty vehicles, he is rejected from the system. However, the system is designed so that such events occur rarely. Specifically, a performance standard is set so that such events occur with likelihood less than or equal to $\tilde{\pi}_0$ for any customer in the steady state. Generally, $\tilde{\pi} := .01$ is the standard. When the standard is so stringently set, whether the customer is rejected or waits at the station until an empty vehicle appears makes little difference for planning purposes. Therefore, rejecting customers, while perhaps not so realistic, is a justifiable assumption for our objective.

This section uses two techniques to model push control. The first technique is discussed in Section 4.6.1 and uses the theory of queueing networks previously introduced in Section 4.4 as a framework to model a full PRT network. First, a simple BCMP model of an uncontrolled network is developed before extensions to push control are considered. The second technique is discussed in Section 4.6.2. It uses a much simpler birth-death framework and focuses only on the distribution of empty vehicles at stations considered in isolation from the rest of the network.

Clearly, the isolated station model is in some sense less accurate than the full network model because it cannot account for the distribution across entire network. Why bother to develop both frameworks?

The benefit of the full network model is that it enables the distribution of vehicles throughout the entire network to be found precisely, accurately, and quickly. However, recall that although the distribution of BCMP networks can be expressed analytically up to a constant $G$, finding (or circumventing) $G$ relies on a fairly opaque, non-trivial numerical procedure. Because of this, optimization of the push parameters (i.e. selecting those parameters that will minimize the fleet size while satisfying the service standard) under the BCMP model must also be done numerically, making global optimization susceptible to the local minima trap as well as making further insights difficult.

Yet, because the BCMP model describes the distribution of the entire network, it is useful as a benchmark against which the accuracy of the simpler isolated station model
can be measured. It will be seen that the isolated station model, despite a “lead cheap” formulation, closely matches the “gold standard” of the full network model. The benefit of this stochastic alchemy is that the isolated station model can be expressed analytically without opaque constants and with the confidence that little is lost in considering a palpably simpler framework.

In particular, this approximation makes optimization of control parameters easier. In fact, the problem will be framed so that all of the control variables are determined by a single underlying decision variable, which expresses empty vehicle flow throughout the network. As a result, in addition to quickly finding the distribution of the network, we will be able to quickly optimize it too.

4.6.1 A Full Network Model of Push Control

This section models push control over an entire PRT network, using the BCMP theory of queueing networks as a framework for analysis. This model will be built incrementally. Before push control is modeled, an uncontrolled network is considered first.

4.6.1.1 A BCMP Model of an Uncontrolled Network

We progress gradually towards a BCMP model of a push control to be presented in the next subsection. To begin, this subsection considers an uncontrolled network, where there is no redistribution of empty vehicles.

The goal here is to find the stationary distribution of vehicles at each station across the entire PRT network. This problem is complicated by the interdependence of stations to one another. At a given time, the number of vehicles at a particular station is not independent of the number vehicles at other stations. At first glance, it might appear that to find the vehicle distribution at one station requires formulating the joint distribution across all possible states over all stations in the network. Yet, the state space would explode for large $n$ or large fleet size, making such a task computationally intractable for even modest networks. However, leveraging the BCMP Theorem as described in Section 4.4 allows the joint distribution to be written as the product form of the distributions of vehicles at each station.

To frame the problem as a BCMP network we employ two node types: one set corresponding to user arrivals at the stations, and the other set relating to trip time generation as the vehicles transport passengers between stations.

At the station nodes, the intuitive customer/server roles are inverted. That is, we consider vehicles as customers and passengers as servers. Now the queue length at a station node corresponds to the number of available vehicles at that station. A “service” at a station node corresponds to a passenger arrival and passengers never wait. Because the arrival process is assumed to be Poisson and passengers never wait, the server is exponential with service rate equal to passenger arrival rate.
Once service at a station node is completed (a passenger arrives), the vehicle is routed to an infinite server node where it again plays the role of the customer. Here, the server corresponds to a generator of the trip time between origin and destination stations. An infinite server queue is appropriate because it is assumed that the network has been designed to accommodate the load with minimal delay, making capacity considerations irrelevant.

Note that the BCMP network is closed since over the course of normal operations vehicles are neither created nor destroyed (one hopes).

Figure 4.6.1.1 depicts the model with three stations, although larger \( n \) could be imagined with \( n(n - 1) \) origin-destination pairs. Here, a passenger arrives at station \( i \) with likelihood \( \lambda_i \), which corresponds to the service rate of the station node. A passenger arrival event (i.e. a completion of service at a station node) sends a vehicle from the station to an origin-destination trip time generator node at random. The routing probabilities correspond to the probability that a passenger arriving at \( i \) is destined for \( j \), which is \( \lambda_{ij}/\lambda_i \). After service is completed at trip generation node \( ij \), the destination is known and the vehicle is routed back to station \( j \) with probability 1.

\[ \pi (\vec{x}) = \frac{1}{G} \prod_{k=1}^{n^2} \pi_k (x_k) \]  

Figure 4.3: BCMP Model of an Uncontrolled Network

The network is comprised of \( n \) single exponential server nodes \( n(n - 1) \) infinite server nodes representing the origin-destination trip generators. By the BCMP theorem the joint distribution across the entire network is the product of the distributions at each of the \( n^2 \) nodes.

When the node represents station \( i \), its steady state distribution is the same as that of an \( M/M/1 \) queue with service rate \( \mu_i = \lambda_i \) and visit ratio \( e_i \) determined by the linear equations as previously described:
\[ \pi_i(x_i) = \left( \frac{e_i}{\lambda_i} \right)^{x_i} \quad (4.6) \]

When the node represents a trip time generator from \( i \) to \( j \), its steady state distribution is the same as that of an infinite server with service rate \( t_{ij} \). This is essentially a Poisson with mean \( e_k t_{ij} \):

\[ \pi_k(x_k) = \frac{1}{x_k!} (e_k t_{ij})^{x_k} \quad (4.7) \]

4.6.1.2 A BCMP Model of a Push-Controlled Network

Push control can be incorporated with a few adjustments to the uncontrolled network model.

Much of the assumptions and parameters from the uncontrolled model carry over to the push control model. When the number of vehicles is below the threshold the service rate reflects demand from station \( i \). If a passenger arrives at station \( i \) and vehicles are available, he travels to destination \( j \) with probability \( \lambda_{ij}/\lambda_i \). If no vehicles are waiting, the passenger is rejected from the system.

The only real difference with push control is that the service rates at station nodes vary according to the number of “customers” awaiting service. (Recall that here customers are actually empty vehicles at each station.) When the number of vehicles at a station is above the threshold, arrivals are processed immediately (i.e. with infinite service rate). More formally the push control service rate is:

\[ \mu_{i,j} = \begin{cases} \lambda_i & j \leq b_i \\ \infty & j > b_i \end{cases} \]

Fortunately, the BCMP theorem permits service rates that depend on queue length. Thus, the joint distribution is the same as that of the uncontrolled network except the distribution at stations is replaced with

\[ \pi_i(x_i) = \frac{(1 - \rho_i) \rho_i^{x_i}}{1 - \rho_i^{b_i + 1}} \]

Here, the \( \rho_i \) are determined by the ratio of the visitation parameters and the service ratio. It is now possible to use Reiser and Lavenberg’s extension to Buzen’s Algorithm to calculate the distribution of available vehicles at each node for a particular fleet size. The results are withheld until the next section when a comparison with the isolated station model is presented.

Under push control, if the number of empty vehicles at station \( i \) exceeds \( b_i \) (a decision variable), then the station immediately releases an empty vehicle to station \( i \oplus 1 \), the station downstream of \( i \) in the TSP loop. However, this cannot be modeled exactly with a BCMP framework, which allows static routing probabilities only. It is not be permissible
for empty vehicles to be routed with one set of probabilities and full vehicles to be routed with another. Therefore, it is assumed that station $i$ pushes vehicles to station $j$ with likelihood $\lambda_{ij}/\lambda_i$.

For our purpose, this is not really an issue. The network model is developed chiefly as a standard to justify the use of a simpler isolated station model. Verification that both models output similar results under static routing of pushed vehicles justifies the use of the isolated station model under different routing behavior. Specifically, we will compare the performance to that of an isolated station model in which vehicles are always pushed from station $i$ to $i \oplus 1$.

### 4.6.2 An Isolated Station Model of Push Control

Here a much simpler model of push control is proposed. Instead of finding the probability distribution of the entire network, we focus on the distribution of vehicles at each station considered in isolation.

This section is organized as follows. First, we discuss the concept of push generated surplus flow, which is the single decision variable driving the optimization of the control. Second, we model isolated stations using a birth-death Markov process. Third, we optimize the flow of empty vehicles to minimize the total number of vehicles required to satisfy the performance standard. Finally, we compare the probability distribution of available vehicles at stations to the distribution found using to the full network model developed in the last section. We will see that little is lost in using the simpler technique developed here.

#### 4.6.2.1 Flow Surplus

Ostensibly, this section considers stations in isolation. Yet there is an inextricable linkage between stations that cannot be completely ignored. How trigger happy a particular station in pushing empty vehicles affects the flow of empty vehicles throughout the rest of the network and the distribution of available vehicles at other stations. This linkage is represented by $\alpha$ which we call *surplus flow*.

Low push thresholds trigger more pushes and thus produce high surplus flow. Likewise, high thresholds result in low surplus flow. In the limit, as the threshold $b \to \infty$, $\alpha \to 0$. While it is more natural in the causal sense to think of the threshold values as determining surplus flow, the optimization procedure that follows regards each $b_i$ as a function of $\alpha$. This allows $\alpha$ to act as a proxy for threshold values. The approach ensures balanced flow at each station while using $\alpha$ as the only decision variable. This makes the optimization convenient and analytically tractable.

As with the previous BCMP model of push control, the isolated station model dispatches empty vehicles from station $i$ when the number of empty vehicles on hand exceeds a threshold $b_i$. However, under the isolated station model, these thresholds are not considered independent decision variables. Rather, each $b_i$ is derived from $\alpha$, which represents
the surplus flow of empty vehicles across the network.

Here “surplus” refers to the additional empty vehicle flow passing through each station above a baseline of empty vehicle flow denoted by the matrix $\hat{\epsilon}$, the solution to the MCNF of Equation 4.2. That is, empty vehicles enter station $i$ at a rate of $\hat{\epsilon}_i + \alpha$ and empty vehicle leave the station at a rate of $\hat{\epsilon}_i + \alpha$. Recall that the MCNF solution guarantees balanced vehicle flows into and out of each station so that $\lambda_i + \hat{\epsilon}_i = \lambda_i + \hat{\epsilon}_i =: \hat{\lambda}_i$. Adding the baseline flow to the surplus flow, the total flow of both empty and full vehicles into and out of each station is $\hat{\lambda}_i + \alpha$.

The surplus empty vehicle flow is channeled along the loop defined by the TSP solution, so that station $i$ pushes excess vehicles to station $i \oplus 1$. It may be that the MCNF solution coincides with the TSP solution at a particular station, directing all empty vehicles flow from $i$ to $i \oplus 1$. This will often be the case, particularly if the network is a simple loop. When this is not the case, though, the routing of pushed vehicles must ensure that the baseline MCNF flows are maintained. To do so, a pushed vehicle is randomly routed to station $i \oplus 1$ with likelihood $(\alpha + \hat{\epsilon}_{i,i\oplus 1}) / (\alpha + \hat{\epsilon}_i)$ and is routed to station $j \neq i \oplus 1$ with likelihood $(\hat{\epsilon}_{i,j}) / (\alpha + \hat{\epsilon}_i)$. Thus, when $\alpha = 0$, we are left with the MCNF flows, and when $\alpha > 0$, surplus flow is channeled through the TSP loop.

### 4.6.2.2 A Birth Death Model of the Isolated Station

From the perspective of a given station $i$, the rest of the network can be thought of as a single node with an infinite buffer to absorb vehicles pushed from station $i$. The rest of the network also pushes vehicles, empty and full, to station $i$ as two independent Poisson processes. Though the push processes of station $i$ and the rest of the network are parametrically linked by $\alpha$, they are assumed temporally independent. Thus, the dynamics of the empty vehicle inventory at a particular station is well modeled by a Markovian birth-death model, as depicted in Figure 4.4.

Let us consider the death and birth rates.

Under push control, the inventory decreases only when a passenger departs the station, which occurs at a rate of $\lambda_i$, i.e. the death rate.

The rest of the network sends full vehicles to be unloaded at destination $i$ with rate $\lambda_i$ and pushes empty vehicles at a rate of $\hat{\epsilon}_i + \alpha$. For the purpose of tracking the inventory of empty vehicles, though, it need only be noted that vehicles arrive at at a rate of $\lambda_i + \hat{\epsilon}_i + \alpha = \hat{\lambda}_i + \alpha$. Thus, until the inventory reaches a level of $b_i$ vehicles, the birth rate is $\hat{\lambda}_i + \alpha$.

When the station has $b_i$ vehicles, it is saturated: arriving vehicles do not cause a change in inventory, but they do trigger a push. So in state $b_i$ the station pushes at a rate of $\hat{\lambda}_i + \alpha$. 

Having formulated a birth-death chain model for push control, we can now find the stationary distribution of empty vehicles at station $i$. Letting $\pi_k$ denote the likelihood of $k$ vehicles, and plugging in the birth-death parameters to Equation 4.1, we have:

$$\pi_k(i) = \pi_0 \left( \frac{\hat{\lambda}_i + \alpha}{\lambda_i} \right)^k \text{ for } 1 \leq k \leq b_i$$

$$\pi_0(i) = \left[ 1 + \sum_{k=1}^{b_i} \left( \frac{\hat{\lambda}_i + \alpha}{\lambda_i} \right)^k \right]^{-1} \quad (4.8)$$

As a matter of notational convenience, the index argument to the stationary distribution $(i)$ will be dropped where its context is obvious.

### 4.6.2.3 Optimization

To reiterate, the central concern of this chapter is to find the smallest number of vehicles needed so that the likelihood of an empty station is at or below the design standard $\tilde{\pi}_0$. Under isolated station push control, the number of vehicles in the network is optimized over a single parameter. The surplus flow $\alpha$ represents a tradeoff between the number of empty vehicles in circulation and idle at stations. Obviously, the larger $\alpha$ is, the more empty vehicles will be in circulation. The smaller $\alpha$ is the larger the push threshold $b_i$ (which can be thought of as a function of $\alpha$) must be to satisfy the performance standard – and the larger $b_i$ is, the more vehicles will be idle at station $i$.

First, each $b_i$ will be formulated as a function of the surplus flow variable $\alpha$, the demand matrix, $\lambda$ and the design standard $\tilde{\pi}_0$. Then, an expression for the total number of vehicles required will be formulated as a function of $\alpha$, $\lambda$, and $t$, enabling a quick and easy optimization over $\alpha$.

Under fleet size optimization, it is obvious that the design standard $\pi_0 \leq \tilde{\pi}_0$ should be a binding constraint. Thus for each $i$ and fixed $\alpha$, we can set $\pi_0 = \tilde{\pi}_0$ in Equation 4.8 and solve for $b_i$ as a function of the other parameters. Honoring the discreteness of the state space, i.e. the number of available vehicles, the solution should be discretized with $\lceil \cdot \rceil$, the ceiling operator:

$$b_i(\alpha, \lambda, \tilde{\pi}_0) = \left\lceil \log \left\{ 1 - \left[ \frac{\pi_0^{-1} - 1}{1 - \left( \frac{\hat{\lambda}_i + \alpha}{\lambda_i} \right)} \right] \right\} \right\rceil - 1 \quad (4.9)$$
This discretization complicates things a bit. Furthermore, while it is trivial to solve for the push threshold as a function of the other parameters under the discrete formulation of push control, the pull control (to be developed in the next section) cannot be expressed quite so tidily. To avoid these complications and ensure consistency, we approximate the state space as continuous. That is, we simply take the discrete probability distribution, and apply it as if the support were continuous. (Most of the math that follows could carry through analytically in the discrete space with nominal changes, but the continuum approximation makes things simpler.) The normalization of Equation 4.8 becomes

\[ \pi_0^{-1} = \int_0^{b_i} \left( \frac{\hat{\lambda}_i + \alpha}{\lambda_i} \right)^x dx = \frac{\left( \frac{\hat{\lambda}_i + \alpha}{\lambda_i} \right)^{b_i} - 1}{\log \left( \frac{\hat{\lambda}_i + \alpha}{\lambda_i} \right)} \]

Fixing \( \pi_0 \) to the design standard \( \bar{\pi}_0 \), and solving for \( b_i \) with the continuum approximation we find:

\[ \tilde{b}_i (\alpha, \lambda, \bar{\pi}_0) = \log \left[ \frac{\log \left( \frac{\hat{\lambda}_i + \alpha}{\lambda_i} \right)}{\bar{\pi}_0} + 1 \right] / \log \left( \frac{\hat{\lambda}_i + \alpha}{\lambda_i} \right) \]

which differs very little with the discretized threshold.

Each vehicle can be classified as being in one of three states: (1) occupied and in transit; (2) empty and in transit; or (3) empty and waiting at a station. Consider the mean number of vehicles in each category under the design standard. The first element is \( \sum \lambda_{ij} t_{ij} \) and can be ignored in the optimization because it is independent of \( \alpha \). The second element is the sum of the mean number of empty vehicles under the baseline flow, \( \sum \hat{\epsilon}_{ij} t_{ij} \), which is constant with respect to \( \alpha \), plus the number of surplus empty vehicles traveling through the TSP loop. This is the product of the flow \( \alpha \) and the distance of the TSP loop, which is defined as \( s \), making the mean number of surplus empty vehicles in motion \( \alpha s \). The likelihood that \( x \) vehicles are waiting at station \( i \) is \( \bar{\pi}_0 \left( \frac{\hat{\lambda}_i + \alpha}{\lambda_i} \right)^x \) and the mean number of empty vehicles waiting at station \( i \) is the integral (not a discrete sum, due to the continuum approximation) of the product of the density and \( x \) over the support space, which is bounded below by 0 and above by the saturation point \( b_i \). Therefore, the fleet minimization problem becomes, (including only the terms involving \( \alpha \)):

\[ \min_{\alpha} \left\{ \alpha s + \sum_{i=1}^{n} \int_0^{b_i} \bar{\pi}_0 \left( \frac{\hat{\lambda}_i + \alpha}{\lambda_i} \right)^x xdx \right\} \]

Evaluating the integral, Equation 4.10 becomes

\[ \min_{\alpha} \left\{ \alpha s + \bar{\pi}_0 \sum_{i=1}^{n} \left[ \frac{b_i}{\log \left( \frac{\hat{\lambda}_i + \alpha}{\lambda_i} \right)} - \frac{1}{\log \left( \frac{\hat{\lambda}_i + \alpha}{\lambda_i} \right)^2} \right] \left( \frac{\hat{\lambda}_i + \alpha}{\lambda_i} \right)^{b_i} + \frac{1}{\log \left( \frac{\hat{\lambda}_i + \alpha}{\lambda_i} \right)^2} \right\} \]
The optimization over $\alpha$ can be done quickly and easily using Newton’s method. Results presenting fleet size as a function of demand will be deferred until they can be compared to pull control, which will be developed in the next section.

To conclude this subsection, it should be noted that alternative, perhaps more realistic assumptions can also be analyzed using the birth-death model. In particular, the BCMP model must assume that arriving passengers who encounter an empty station are immediately rejected from the system. While this does not have a tangible effect on performance for a sufficiently stringent performance standard, it does not consider the somewhat more realistic possibility that passengers queue at the station for vehicles to appear. This could be modeled easily with the birth and death framework by allowing the state space to extend to negative integers corresponding to passengers waiting at the station. This would also allow a service standard based on delay rather than the likelihood of an empty station. However, we do not investigate the delay metric since it would break consistency with the full network model\textsuperscript{2}. The underlying method would be quite similar in either case.

4.6.3 Isolated Station and Full Network Models Compared

Here, we compare the stationary distribution of available vehicles under the BCMP and isolated station models. Each model is applied to the simple homogenous loop configuration described in Section 4.2.3.1. Without loss of generality, the loop travel time is unity. Total demand $\Lambda$ is fixed. Figure 4.5 compares the distribution of available vehicles at a station that is obtained with the methods of Section 4.6.1 and Section 4.6.2 for four values of $n$, the number of stations.

For comparison, the same optimal threshold variable $\tilde{b}_i(n)$ and fleet size $\tilde{f}(n)$ from the isolated station model was used for the BCMP model.

\textsuperscript{2}Waiting passengers might also be modeled on the full network level by use of Gelenbe networks, which extend BCMP networks to allow so-called negative customers to be admitted to the network. However, the complexity involved in the modeling and computation of such distributions is considerably greater than that of simple BCMP networks, and so their utilization is reserved for future work.
Figure 4.5: Stationary Distribution of Available Vehicles Under Push Control Model

Figure 4.5 shows that the distributions of the BCMP network and the isolated station model track each other very closely. There is a noticeable discrepancy for small $n$ but as the number of stations increases, the distributions are nearly identical. (Note also that as $n$ increases, the push threshold decreases because the smaller inter-station travel times allow a more aggressive pushing.) This justifies the isolated station model as an excellent approximation of the larger network for all but the smallest networks.

### 4.7 Pull Control

This section considers a strategy called pull control, which is, in a sense, complementary to push control. Under pull control a station requests a vehicle when its inventory falls below a lower threshold. We develop an isolated station model of pull control using a method similar to that of the previous section (allowing this section to proceed at a faster pace). When compared, it will be seen that pull control is generally superior to push control, requiring significantly fewer vehicles to satisfy the performance metric.

Under pull control, station $i$ requests a vehicle when its inventory falls below a lower threshold, denoted by $a_i$. Inventory at station $i$ is decremented when a trip with origin $i$ is initiated or when an empty vehicle is pulled from station $i$ itself. Either type of
vehicle request prompts a pull event if the inventory is below $a_i$. A pull triggers the movement of an empty vehicle from a station (usually, but not necessarily, from station $i \oplus 1$ as explained later) to station $i$. Pull requests do not immediately bring the inventory level back to $a_i$ because the pulled empty vehicle takes some time to reach station $i$. If the inventory at station $i$ is decremented before level $a_i$ is reached a new pull event is initiated. So, if station $i$ has $a_i - x$ vehicles in inventory, it can be inferred that $x$ vehicles are currently en-route to the station.

The queuing network techniques used to model the full PRT network under push control do not apply to pull control. However, if results from the push control are any indication, we may assume that the global performance of a network under pull control will be well approximated by extrapolating the local behavior of isolated stations, provided the number of stations is sufficiently large.

As with push control, the local vehicle inventory at a given station is modeled as a birth-death process. Analyzing station $i$ in isolation, the interaction with the larger network is encapsulated by a small set of parameters.

Similar to push control, we introduce a single global decision variable $\alpha$ symbolizing surplus empty vehicle flow. It acts as a proxy for the pull threshold $a$. High pull thresholds trigger more empty vehicle movement, resulting in high values of $\alpha$. Likewise, low pull thresholds result in less vehicle movement so that $\alpha = 0$ implies that $\alpha = 0$. The optimization reverses the causal intuition, regarding $a_i$ a function of $\alpha$ for each $i$.

The surplus flow represents the flow of pulled empty vehicles beyond the minimum baseline flow as determined by the solution to the MCNF of Equation 4.12 denoted by the matrix $\hat{\epsilon}$. That is, under the baseline MCNF flow, empty vehicles are pulled to and from station $i$ with rates $\hat{\epsilon}_i$ and $\hat{\epsilon}_\cdot_i$ respectively. Including surplus flow, the total empty vehicle flow rates become $\hat{\epsilon}_i + \alpha$ and $\hat{\epsilon}_\cdot_i + \alpha$.

As with push control, the surplus empty vehicle flow is channeled along the loop defined by the TSP solution so that station $i$ requests vehicles from station $i \oplus 1$. If the MCNF and TSP loop do not coincide, vehicles are routed randomly. A pull request from station $i$ pulls a vehicle from station $i \oplus 1$ with probability $\frac{\alpha}{\alpha + \hat{\epsilon}_i}$ and pulls a vehicle from station $j \neq i \oplus 1$ with probability $\frac{\hat{\epsilon}_j}{\alpha + \hat{\epsilon}_i}$. When $\alpha = 0$, the solution reverts to that of the baseline MCNF flow.

The mean travel time of pulled empty vehicles to reach station $i$ will be relevant in formulating the model. This value is denoted by $s_i$ and is equal to the sum of the travel time of vehicles pulled from the station $i \ominus 1$ and the mean travel time of vehicles pulled from the baseline MCNF solution, weighted by the likelihood of each pull event:

$$s_i (\alpha, \hat{\epsilon}, t) = \frac{\alpha}{\alpha + \hat{\epsilon}_i} t_{i \ominus 1, i} + \frac{1}{\alpha + \hat{\epsilon}_i} \sum_{j=1}^{n} \hat{\epsilon}_{ji} t_{ji}$$

A birth-death chain is used to model the dynamics of pull control and is depicted in Figure 4.6.

The death rate reflects inventory depletion events: when either a full vehicle departs.
from origin $i$ or when an empty vehicle is pulled from $i$ by the rest of the network. The former event occurs with rate $\lambda_i$ and the latter with rate $\hat{\epsilon}_i + \alpha$. Recall that by definition $\hat{\lambda}_i := \lambda_i + \hat{\epsilon}_i$. So the combined death rate is $\hat{\lambda}_i + \alpha$, independent of inventory.

By contrast, the birth rate depends on the state of the inventory of available vehicles at station $i$. Two processes increment inventory, full vehicle arrivals and fulfilled pulled requests. The birth rate is the sum of the rates of the two processes.

First, trips ending at destination $i$ contribute $\lambda_i \cdot i$ to the birth rate, independent of the state of the inventory at station $i$. The second contributor to the birth rate is the pull process, which does depend on the state of the inventory.

When the number of vehicle in stock is $k < a_i$, we know that $a_i - k$ vehicles are currently en-route to station $i$. Although inter-station travel times are fairly deterministic, they are assumed to be exponential random variable with mean $s_i$ as calculated by Equation 4.12. The distribution should have a minor effect since the stationary distribution of the inventory is mostly affected by the mean replenishment rate.

With $k < a_i$ vehicles in inventory, the time until the next pulled vehicle reaches station $i$ is the minimum of $a_i - k$ independent exponential random variables, each with mean $s_i$. The minimum of exponentials is itself an exponential with mean $s_i / (a_i - k)$. The contribution to the birth rate, is the reciprocal.

Adding the rate of trips ending at station $i$ to the pull rate, the birth rate as a function of the inventory state is thus

$$\ell_k = \begin{cases} \lambda_i + (a_i - k) / s_i & \text{for } 0 \leq k < a_i \\ \lambda_i & \text{for } k \geq a_i \end{cases}$$

The stationary distribution is found using the same flow balance principle employed with push control. The only change is that the state space is unbounded on the right.

$$\pi_k (i) = \begin{cases} \pi_0 \prod_{j=1}^{k} \frac{\lambda_i + (a_i - k) / s_i}{\lambda_i + \alpha} & \text{for } 1 \leq k \leq a_i \\ \pi_0 \left( \prod_{j=1}^{a_i} \frac{\lambda_i + (a_i - k) / s_i}{\lambda_i + \alpha} \right) \left( \frac{\lambda_i}{\lambda_i + \alpha} \right)^{k-a_i} & \text{for } k > a_i \end{cases}$$

$$\pi_0 (i) = \left[ 1 + \left( \frac{a_i}{\lambda_i} \right) \prod_{j=1}^{k} \frac{\lambda_i + (a_i - k) / s_i}{\lambda_i + \alpha} + \left( \frac{a_i}{\lambda_i} \right) \prod_{j=1}^{a_i} \frac{\lambda_i + (a_i - k) / s_i}{\lambda_i + \alpha} \right] \sum_{k=a_i}^{\infty} \left( \frac{\lambda_i}{\lambda_i + \alpha} \right)^{k-a_i} \right]^{-1}$$
Note that the infinite sum is a geometric series so that the equations above can be expressed in closed form. Unfortunately though, unlike push control, it is not possible to isolate the threshold parameter, \( a_i \), as a function of the other parameters. However, it is computationally trivial to simply increment \( a_i \) until \( \pi_0 (a_i, \alpha, \lambda, t) < \tilde{\pi}_0 \) thereby satisfying the design standard. (Recall that \( s_i, \lambda_i \) and \( \hat{\lambda}_i \) are derived from the matrices \( \lambda \) and \( t \).) Denote the minimum value of \( a_i \) for which the design standard inequality is satisfied as \( \tilde{a}_i (\alpha, \lambda, t, \tilde{\pi}_0) \) or \( \tilde{a}_i (\alpha) \) with the other arguments are implied.

In this section, we have developed a procedure to find the pull control threshold for a given network and flow variable \( \alpha \). We can now minimize the fleet size over the single parameter \( \alpha \). For a given network, we employ a line search algorithm over \( \alpha \) to find the value that minimizes the total fleet size, which is formulated as the expected value of the sum of the surplus vehicles and the number of vehicles at each station:

\[
\min_{\alpha} \left\{ \alpha s + \sum_{i=1}^{n} \left[ \hat{a}_i(\alpha) \sum_{k=1}^{\infty} k \pi_k + \pi_{\tilde{a}_i} \sum_{k=\tilde{a}_i}^{\infty} k \left( \frac{\lambda_i}{\lambda_i + \alpha} \right)^{k-\tilde{a}_i} \right] \right\}
\]

Finally, with procedures in place to find the minimum fleet size under both push and pull control methods, we can evaluate each controller. Figure 4.7 plots fleet size over a homogenous network with \( n = 16 \) stations over a range of total demand \( \Lambda \). Four curves are plotted. Two of the curves show benchmarks as described in Section 4.5. Fleet decentralization provides an upper-bound for required fleet size and empty vehicle “teleportation” provides a lower bound on fleet size. The other two curves depict the minimized fleet size under optimal push control and pull control. Inspired by the forms of the benchmarks, these two curves are actually the fleet sizes regressed on the sum of demand, the square root and an intercept. That is, push and pull were each fit to according the least squares regression

\[
f = c_1 \Lambda + c_2 \sqrt{\Lambda} + c_3
\]

for linear model coefficients \( c_1, c_2, \) and \( c_3 \). Both the push and pull data fit this linear model extremely closely with \( R^2 \) values exceeding .987. As might have been expected, this suggests that required PRT fleet size scales with demand in the same manner as the number of servers needed to operate standard queueing systems.
As Figure 4.7 shows, pull performs significantly better than push control. It should also be noted that the fleet size for both control policies hems closer to the lower bound than the upper bound.
Chapter 5

Conclusion
Three factors can undermine the performance of a particular personal rapid transit system: congestion on guideways, congestion to and from stations, and scarcity of empty vehicles at stations. In this thesis, each problem has been allotted a chapter.

Chapter 2 focused on a potential source of congestion on guideways, namely merges. The analysis rested on a novel yet simple decomposition of the merge into three deterministic server blocks. This framework enabled the rediscovery of a previously known formula for mean merge delay under a first-come-first-merge rule. Extending the approach, we were able to find a new formula bounding the variance of merge delay. This allowed us to construct a bound on the buffer length required for merge approaches so that buffer overflow is rare. The result can aid planners in determining whether guideways have sufficient capacity to handle vehicle flows so that delay and overflow likelihoods fall below certain design standards.

Chapter 3 examined stations. Here, we compared the performance of two archetypal station configurations. Under the parallel configuration, vehicles can freely enter and exit berths independent of other vehicles. Under the serial configuration, each stopped vehicle at a berth blocks upstream vehicles from exiting or entering downstream berths. Not surprisingly, parallel stations performed much better, incurring less delay and requiring fewer berths to meet a performance standard. We can conclude that serial stations should be used for very low demand stations only. If it is costly to build parallel or operate stations, a PRT system will not scale well.

Chapter 4 developed tools for fleet planning and operations at the network level. The questions addressed were: How should empty vehicles be dispatched? and Given an empty vehicle controller, how many vehicles are required to meet a performance standard? The control policies examined were so-called push and pull control. To study push control, a framework based on queueing networks produced the joint distribution across all stations. Yet, examining stations in isolation from the rest of the network produced very similar results. This also justified an isolated station model of pull control, which was found to perform significantly better than push control in the sense that it requires fewer vehicles to achieve the same service standard.

Although the topics from each chapter are quite distinct, the underlying methods used to explore them were similar. Whether a vehicle passes through a merge, a station, or the broader network, each problem involved the analysis of the flow of objects in a network that contains “one or more locations at which there is some restriction on the times or frequencies at which the objects can pass” to quote Gordon Newell’s definition of a queueing network.[27] In the case of the merge, safety considerations limit the frequency of vehicles that can pass through a particular point on the guideway. For stations, the time to load and unload vehicles limits the availability of berths. Over the broader network, passenger demand limits the frequency of vehicle trips and the distance between stations restricts the redistribution time.

It is therefore not surprising that for each of the three topics, queues provided a convenient abstraction, the specific methodology tailored to the problem. Merges were represented as decomposition of a series of M/D/1 queues. Station behavior was depicted
as a Markovian queueing process. Finally, vehicle movement through the network was framed as a BCMP queuing network.

While such abstractions necessarily omitted details that could be included in more intricate simulations, the advantage was access to a well-stocked toolbox of analytical and numerical methods developed over many decades of queueing theory. This allowed us speed and insight lacking in simulations, which currently comprise the bulk of existing PRT literature.

This thesis has taken several small but significant steps in characterizing and planning personal rapid transit systems. The work has furthered an understanding of the capabilities and limitations of a PRT system over microscopic and macroscopic levels. Planners and operators may find the analytical and numerical methods described in this thesis useful as a first pass in understanding the requirements of a particular network. However, planners will likely need to use simulations at more refined stages of the planning process. Extending the models developed here to allow finer details would be helpful in shrinking the territory ruled by simulation.

A number of areas warrant future exploration.

Each previous chapters assumed time invariant demand to plan for the worst-case of sustained peak demand. But real-word systems might exhibit bursty demand. For example, airport PRT systems may handle brief surges of demand when planes unload. For such a circumstance planning based on the peak demand would be overly cautious. Extending the models developed in each chapter to incorporate time-varying demand should be useful in more realistic analyses of merges, stations, and fleet operations.

There are also topics for future research specific to each domain discussed in this thesis.

The merge analysis assumed that the demand rates entering each merge point were fixed. However, the central controller could route vehicles to minimize the total passenger delay. The merge delay can provide the building block to formulate system-wide delay parametrically. This might facilitate calculation of system-optimal routing for PRT systems.

The station analysis compared two station configurations that could almost be described as caricatures. Actual PRT stations would likely be a hybrid of both serial and parallel berth stations. Allowing for more complex configurations could be a rich area for future research. Additionally, vehicle-following dynamics will affect the stations capacity and operations in ways that cannot be modeled by the simple server model. A side-by-side comparison of a queueing model to a simulation of the lower level physics would help assess the significance of such details. Finally, a more comprehensive approach to station control would be salient to real-world operations.

While we were able to model push control as a queueing network, we were unable to do so for pull control. We are confident that the accuracy of the isolated station model found under push control applies to pull control. However, it would be reassuring to compare the isolated station model of pull control to a full network model. In the last two decades a profusion of theoretical queueing literature has broadened the class of product-form networks beyond BCMP networks. It is quite possible that one of these
extensions might support a pull control model. In particular, Gelenbe networks look like a hospitable framework to pull control. Furthermore, a natural extension of the push and pull controls would be a hybrid push-pull control, where each station has two control parameters: a lower and an upper threshold that trigger pulls and pushes respectively. Clearly, such an approach would achieve superior performance to exclusive push or pull control.

Every mode of transportation brings with it an experience for users and nonusers alike that cannot be measured solely in terms of how efficiently passengers are transported from point A to point B (or \(i\) to \(j\)). The social implications of individual versus group transit, the effect on land use patterns, and the aesthetics of infrastructure are just a few of the many issues that resist quantitative analysis. Ultimately, the merit of PRT systems rest on factors like these as much as technical considerations.
Bibliography


