1-1-2012

Transit Service Reliability: Analyzing Automatic Vehicle Location

CUTR

Follow this and additional works at: https://digitalcommons.usf.edu/cutr_nctr

Recommended Citation

"Transit Service Reliability: Analyzing Automatic Vehicle Location," National Center for Transit Research (NCTR) Report No. CUTR-NCTR-RR-2012-15, Center for Urban Transportation Research, University of South Florida, 2012.

DOI: <https://doi.org/10.5038/CUTR-NCTR-RR-2012-15>

Available at: https://scholarcommons.usf.edu/cutr_nctr/88

This Technical Report is brought to you for free and open access by the National Center for Transit Research (NCTR) Archive (2000-2020) at Digital Commons @ University of South Florida. It has been accepted for inclusion in Research Reports by an authorized administrator of Digital Commons @ University of South Florida. For more information, please contact digitalcommons@usf.edu.

**Transit Service Reliability:
Analyzing Automatic Vehicle Location (AVL) Data
for On-Time Performance and Identifying
Conditions Leading to Service Degradation**

Final Report

November 2016

PROJECT NO.
2117905017

PREPARED FOR
National Center for Transit Research

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation University Transportation Centers Program and the Florida Department of Transportation, in the interest of information exchange. The U.S. Government and the Florida Department of Transportation assume no liability for the contents or use thereof.

The opinions, findings, and conclusions expressed in this publication are those of the authors and not necessarily those of the State of Florida Department of Transportation.

Transit Service Reliability

Analyzing Automatic Vehicle Location (AVL) Data for On-Time Performance and Identifying Conditions Leading to Service Degradation

Submitted by:

Fabian Cevallos, Ph.D.
Transit Program Director
Lehman Center for Transportation Research (LCTR)
Florida International University (FIU)
10555 West Flagler Street, EC 3609
Miami, FL 33174
Phone: (305) 348-3144
Email: fabian.cevallos@fiu.edu

Submitted to:

Joel Volinski
NCTR Program Director
National Center for Transit Research (NCTR)
University of South Florida (USF)
4202 E. Fowler Ave., CUT100
Tampa, FL 33620-5375
Phone: (813) 974-9847
Email: volinski@cutr.usf.edu

November 2016

Metric Conversion

SYMBOL	WHEN YOU KNOW	MULTIPLY BY	TO FIND	SYMBOL
LENGTH				
in	inches	25.4	millimeters	mm
ft.	feet	0.305	meters	m
yd.	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
VOLUME				
fl. oz.	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft ³	cubic feet	0.028	cubic meters	m ³
yd ³	cubic yards	0.765	cubic meters	m ³
NOTE: volumes greater than 1000 L shall be shown in m ³				
MASS				
oz.	ounces	28.35	grams	g
lb.	pounds	0.454	kilograms	kg
T	Short tons (2000 lb.)	0.907	megagrams (or "metric ton")	Mg (or "t")
TEMPERATURE (exact degrees)				
°F	Fahrenheit	$\frac{5}{9} (F - 32)$ or $(F - 32) / 1.8$	Celsius	°C

Technical Report Documentation

1. Report No. 2117905017		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Transit Service Reliability: Analyzing Automatic Vehicle Location (AVL) Data for On-Time Performance and Identifying Conditions Leading to Service Degradation			5. Report Date November 2016		
6. Performing Organization Code					
7. Author(s) Fabian Cevallos, PhD Transit Program Director Lehman Center for Transit Research (LCTR) Florida International University (FIU) 10555 West Flagler Street, EC 3609 Miami, FL 33174			8. Performing Organization Report No. U.S.DOT DTRS98-G-0032		
9. Performing Organization Name and Address National Center for Transit Research Center for Urban Transportation Research (CUTR) University of South Florida 4202 East Fowler Avenue, CUT100 Tampa, FL 33620-5375			10. Work Unit No. (TRAIS)		
11. Contract or Grant No.					
12. Sponsoring Agency Name and Address Research and Innovative Technology Administration U.S. Department of Transportation Mail Code RDT-30, 1200 New Jersey Ave SE, Room E33 Washington, DC 20590-0001			13. Type of Report and Period Covered		
14. Sponsoring Agency Code					
15. Supplementary Notes					
16. Abstract Transit agencies are increasingly using AVL data to support key performance indicators, for responding to service complaints, or for reviewing and improving the quality of service. The main objective of this study was to investigate the use of AVL data for improving transit service reliability: 1) improving on-time performance, and 2) monitoring bus schedule adherence and bus bunching. Static and real-time General Transit Feed Specification (GTFS) data from HART and data from the MDT AVL system were used.					
17. Key Words			18. Distribution Statement No restrictions		
19. Security Classification (of this report) Unclassified	20. Security Classification (of this page) Unclassified	21. No. of Pages		22. Price	

TABLE OF CONTENTS

ABSTRACT.....	1
BACKGROUND	2
LITERATURE REVIEW	4
Definition of Service Reliability	4
On-time Performance (OTP).....	5
Limitation of Traditional Performance Indicators	6
Possible Causes of Service Unreliability	7
Restoration Strategies.....	11
Literature Review Summary	15
ASSESSMENT OF CAD/AVL SYSTEMS	18
Innovative Use of AVL Data	18
Transit Operations Decision Support Systems (TODSS)	21
Computer-Aided Dispatch Systems with Transit Quality of Service Alerts.....	22
TODSS Architecture and Concept of Operations	22
STUDY APPROACH.....	26
General Transit Feed Specification (GTFS).....	26
Creating a GTFS Static Feed.....	27
Make a Transit Feed Publicly Available	27
Submit a Transit Feed to Google	27
Steps for Adding Transit Feed to Google Maps	28
Host a Feed	28
Validate the Data Feed:	29
Feed File Requirements	29

Process and Publish the Existing Feeds by Transit Agencies	30
Understanding GTFS-Realtime	32
Overview of GTFS-Realtime Feed Types	32
Data Format	32
Data Structure	33
GTFS-realtime Term Definition	33
Important Feed Elements	33
Non-GTFS Feeds Published by Transit Agencies	35
Combining GTFS Real-Time and Static Data	36
Tri-Met Developer Resources	37
Other Considerations	44
GPS Coordinates: Latitude and Longitude	44
Software Development	47
CASE STUDIES	53
On-Time Performance Analysis	55
Data preparation	55
Fitting Distributions	57
Bus Bunching Analysis	74
Prevention of Bus Bunching	74
Calculation of Bus Bunching	75
PRODUCING ON-TIME PERFORMANCE FROM GTFS-REALTIME DATA	87
GTFS-realtime – The Raw Data	87
Vehicle Positions	87
Trip Updates	88
Producing Schedule Deviation from GTFS-realtime Data	88
Determining When a Vehicle Arrived at a Stop	89
Proof-of-Concept Open-Source Software Tool	92
CONCLUSIONS, OBSERVATIONS, AND NEXT STEPS	94
Results from Case Studies	96
Prevention of Bus Bunching	96
Calculation of Bus Bunching	97

General Observations	98
Data Issues	98
Departmental Considerations	101
Next Steps	101
REFERENCES	102
APPENDIX A: General Transit Feed Specification Reference.....	109

LIST OF FIGURES

Figure 1: Transit Unreliability Framework.....	8
Figure 2: Identification of Bunching using a Time-Space Diagram.....	9
Figure 3: Proposed Planning Tools.....	12
Figure 4: Determination of Optimal Bus Running and Layover Times	13
Figure 5: Concept of Operations TODSS System Design.....	24
Figure 6: Transit Data in Google Maps	28
Figure 7: Google Transit Data Feed.....	30
Figure 8: GTFS Data Exchange.....	31
Figure 9: Agencies with Real-Time Feeds.....	35
Figure 10: Static and Real-Time GTFS Data Diagram.....	36
Figure 11: Tri-Met Developer Resources	39
Figure 12: Tri-Met's Registration Page	40
Figure 13: Screenshot of Tri-Met XML Data Responses	41
Figure 14: Process of Storing Tri-Met XML Data Responses.....	42
Figure 15: Screenshot of Saving Tri-Met XML Data Responses to SQL Server.....	44
Figure 16: Latitude.....	45
Figure 17: Longitude.....	45
Figure 18: Route Patterns.....	47
Figure 19: Software Development Life Cycle (SDLC)	48
Figure 20: Architectural Overview of a Reliability System	50
Figure 21: Use Case	51
Figure 22: N-Tier Architecture	52
Figure 23: HART Transit Route 5	53
Figure 24: Miami-Dade Transit Route 288.....	54
Figure 25: Miami-Dade Transit Route 11.....	54
Figure 26: Scheduled Adherence Histogram for Route 11	57
Figure 27: Normal Distribution Fitted on Route 11 Schedule Adherence Histogram.....	58
Figure 28: Smallest Extreme Value Distribution Fitted on Route 11 Data	59
Figure 29: Largest Extreme Value Distribution Fitted on Route 11 Data	59
Figure 30: 3-Parameter Lognormal Distribution Fitted on Route 11 Data.....	60
Figure 31: Smallest Extreme Value Distribution Fitted on Route 11 (EB) Data.....	62
Figure 32: Smallest Extreme Value Distribution Fitted on Route 11 (WB) Data	63
Figure 33: Smallest Extreme Value Distribution Fitted on Trip 3491000.....	64
Figure 34: Smallest Extreme Value Distribution Fitted on Adjusted Trip 3491000	66
Figure 35: Before/After Adjustment Distribution Plots for Trip 3491000	67
Figure 36: Before and After Adjustment at the Timepoint Level.....	68
Figure 37: Before/After Distributions with Variability Adjustments for Trip 3491000	68
Figure 38: Smallest Extreme Value Distribution Fitted on Adjusted Trip 3491108	70

Figure 39: Before/After Adjustment Distribution Plots for Trip 3491108	71
Figure 40: Before and After Adjustment at the Timepoint Level.....	72
Figure 41: Before/After Adjustment of Scale Parameter for Trip 3491108	73
Figure 42: Before/After Adjustment of Location Parameter for Trip 3491108.....	74
Figure 43: Static and Real-Time GTFS Tables.....	76
Figure 44: Dataset Results for Route 5	77
Figure 45: Dataset for Individual Trips	78
Figure 46: Distance and Cumulative Distance.....	79
Figure 47: Timestamp and Distance Data.....	80
Figure 48: Hart Transit Route 5	81
Figure 49: Hart Transit Route 5	81
Figure 50: Hart Transit Route 5	82
Figure 51: HART Transit Route 5 (Trip #42359 Southbound)	83
Figure 52: Miami-Dade Transit Route 288.....	84
Figure 53: Miami-Dade Transit Route 288 (Trip #3502699 Eastbound)	85
Figure 54: Trips by Time of Day on SW 88 Street/SW 127 Avenue	86
Figure 55: Example GPS Data and Closest Stops for a Vehicle Running a Transit Route	90
Figure 56: The On-Time Performance Calculator Open-source Tool	92

LIST OF TABLES

Table 1: Transit Level of Service.....	5
Table 2: Headway Coefficient of Variation.....	6
Table 3: Possible Causes of Service Variability in Public Transport	7
Table 4: Possible Causes for Late Arrivals and Early Departures.....	11
Table 5: Strategies on Service Reliability.....	12
Table 6: Agencies' Operational Strategies	15
Table 7: Planning and Real-Time Operational Strategies.....	17
Table 8: CUMTD Uses of AVL and APC Data	20
Table 9: Feed Header and Entity	34
Table 10: Example of GTFS Field Information.....	34
Table 11: Tri-Met's Unofficial GTFS Data Elements.....	37
Table 12: Commonly Used Transit Terms Used in Tri-Met's Data and their Definitions.....	38
Table 13: Tri-Met Web Services.....	41
Table 14: Box Plot Method Applied to Remove Outliers (Adherence Variable).....	56
Table 15: Distributions Analysis for Route 11	57
Table 16: On-Time Performance Measure Percentile for Two Sample Trips	65
Table 17: On-Time Performance Value Adjustment.....	65
Table 18: Adjustment Value for Existing Adherence in Trip 3491000.....	66
Table 19: Before/After Adjustment Data Statistical Parameters	67
Table 20: On-Time Performance for Trip 3491108.....	69
Table 21: On-Time Performance Value Adjustment	69
Table 22: Adjustment Value for Existing Adherence in Trip 3491108.....	70
Table 23: Before/After Adjustment Data Statistical Parameters	71
Table 24: Calculating Simplified Schedule Deviation from GPS Data.....	91

ACKNOWLEDGMENT

This research was managed and funded by the National Center for Transportation Research (NCTR), under the direction of Mr. Joel Volinski. We would like to thank Dr. Sean Barbeau from the Center for Urban Transportation Research (CUTR) for his reviews and valuable contributions to this project, including authoring the chapter on “Producing On-time Performance from GTFS-realtime Data” and leading the development of the On-Time Performance Calculator open-source software tool (<https://github.com/CUTR-at-USF/ontime-performance-calculator>). We would also like to thank Dr. Kirolos Haleem for his contributions and early data analysis.

Special thanks go to Thalia Pickering, Research Assistant, and Kollol Shams, Mohamad Rezaei, and Pankaj Mankotia, graduate students in the Lehman Center for Transportation Research at Florida International University (FIU), for assisting with this research. Their contributions are gratefully acknowledged. The proofreading of the final report by Natalie Defraene, Editor, is also appreciated.

The project team would also like to thank Miami-Dade Transit (MDT) and the Hillsborough Area Regional Transportation Authority (HART) in Tampa, Florida for the data used in this project, and Justin Begley from HART, who wrote the initial Assessment of the AVL Systems section and provided support for this study. The project team would also like to thank Dr. Fabian Cevallos, Principal Investigator, for his direction and management of this project.

ABSTRACT

Transit agencies are increasingly using AVL data to support key performance indicators, for responding to service complaints, or for reviewing and improving the quality of service. The main objective of this study was to investigate the use of AVL data for improving transit service reliability: 1) improving on-time performance, and 2) monitoring bus schedule adherence and bus bunching. Static and real-time General Transit Feed Specification (GTFS) data from HART and data from the MDT AVL system were used.

The “Case Studies” section presents techniques to improve service reliability. For instance, in the case of Trip 3491108, a series of steps are presented to improve on-time performance by 1) adjusting the times in the timetable (7% improvement), 2) controlling the variability (44% improvement), and 3) shifting the times in the timetable (2.71% improvement). Different statistical distributions are considered in the calculations from which the Smallest Extreme Value distribution performed the best for the datasets used in this project.

The “Producing On-Time Performance from GTFS-realtime Data” section discusses the opportunities presented by the available of real-time data from many transit agencies in a standardized format, and presents some of the challenges encountered when generating schedule deviation and on-time performance from GTFS and GTFS-realtime data. A proof-of-concept open-source software tool was created to demonstrate a method of calculating a simplified schedule deviation measurement from raw GTFS and GTFS-realtime data.

In public transportation, bus bunching is a condition that results in unreliable service, long wait times for passengers, overcrowded vehicles, and near empty buses. In this report, time-space diagrams and plots are presented to determine bunching, bus locations, and visualize segments that may need special attention.

Areas identified for future research include the standardization of OTP parameters, measuring reliability from the users versus the transit agency perspective, further development of the software tool to produce more accurate schedule deviation and on-time performance measurements, creation of a software tool to assist agencies in adjusting their schedules and reacting in real-time to reduce the occurrence of service degradation, and the role of Planning, Scheduling, and Operations in improving transit reliability.

BACKGROUND

Automatic Vehicle Location (AVL) systems are computer-based vehicle tracking systems that function by measuring the real-time position of each vehicle and relaying this information back to a central location. AVL systems are most frequently used for fleet management to identify the location of vehicles for a variety of purposes, including improved dispatch, operation efficiency, and faster response times to disruptions in service, such as vehicle failure or unexpected congestion; quicker responses to threats of criminal activity; and improved data for future planning, scheduling, or operation purposes.

An AVL system tracks vehicles by using one of the following location technologies: GPS, signpost and odometer interpolation, ground-based radio (Loran-C), or dead reckoning. Out of all of these technologies, GPS is the most effective tracking system for transit because it is the most accurate, it can be used with variable routing and scheduling, and it does not require purchase, installation, or maintenance of wayside equipment. The GPS system works through a network of orbiting satellites that transmit signals to the ground. Special receivers on each vehicle read the available signals and triangulate to determine their position. The geographic location, along with the date, time, and other operational data, is then sent to the transit agency. Data are transmitted at established polling intervals to the transit center with the use of radio or cellular communication and can be used immediately for daily operations or archived for further analysis.

Transit agencies are increasingly collecting and analyzing AVL data to support a variety of operations, scheduling, and service planning activities. Data can be used as a source for key performance indicators, for responding to service complaints, or for reviewing and improving the quality of service. The use of historical data can help identify problems that occurred in the past. Furthermore, analysts can identify recurring problems and develop solutions to these problems.

Data from an AVL system can also be used to measure, monitor, and improve service reliability, also known as on-time performance (OTP). Schedule adherence is a matter of service quality to transit passengers. Therefore, from the service provider's perspective, schedule adherence reflects the quality of the service plan, which directly affects schedules and operations control. It is also important to clarify the difference between schedule adherence and on-time performance. Schedule adherence refers to the difference between real-time and scheduled times of arrival or departures times, usually presented in minutes. On-time performance, on the other hand, is a percentage value used to indicate whether buses arrive or depart late, or are on time or early. Depending on the different technologies, on-time performance can be calculated using arrival, departure, or possibly a combination of both factors.

As on-time performance is important to the transit customer, OTP strategies can improve customer satisfaction and attract new transit riders. Reliability is one of the areas that transit agencies can use to improve service at a relatively low cost. Therefore, the benefit-to-cost ratio of improving on-time performance is expected to be significant.

To improve transit service reliability, there is also a need for a systematic review of historical AVL data in order to identify recurring service problems, as well as to see if there are conditions in the data that exist in the time period preceding the service problem. Early identification of these conditions can help transit agencies make intelligent decisions and determine the best course of action for avoiding service degradation. This can help enhance the quality of service and customer satisfaction.

The main objective of this study is to investigate the use of AVL data for improving transit service reliability. This is achieved by using better on-time performance techniques and by identifying conditions leading to service degradation that can assist transit agencies in providing a higher quality of service. Therefore, the research focuses on two related areas: 1) improving on-time performance, and 2) monitoring bus schedule adherence and bus bunching to prevent service degradation.

LITERATURE REVIEW

This section focuses on identifying research conducted on the analysis of service reliability using Automatic Vehicle Location (AVL). Initial studies mainly concentrate on the configuration of AVL systems and the potential use of data to analyze transit performance. Later studies focus on developing methodologies for analyzing service reliability and strategies to restore services. Unfortunately, most of these analyses are limited to historical archived data, and with the exception of a few studies, findings are more useful for planning rather than operations. Nevertheless, transit agencies have an urgent need for a system that will not only facilitate the decision-making process from a planning perspective, but also for the real-time operations.

The remainder of this chapter is composed of several parts. The first part introduces the definition of service reliability and different measures; the next part discusses on-time performance, limitations of performance indicators and the possible causes of service unreliability and the methodology used in various studies to identify these causes. The last two parts review different restoration measures from real-time and planning perspectives and the innovation of using data from AVL systems. Finally, a summary of the literature review is presented.

Definition of Service Reliability

One of the main concerns of transit agencies is service unreliability. Therefore, agencies attempt to understand the issues associated with service reliability, as well as find ways to measure reliability and improve it.

Previous literature reviews define “service reliability” in different ways. For example, in Transit Cooperative Research Program (TCRP) Report 165, transit service reliability is defined as a measure to determine how close services are to a standard. Abkowitz (1978) defines this in terms of variability of service attributes and its effect on traveler behavior and agency performance. Turnquist (1981) emphasizes the ability of the transit system to adhere to the schedule, maintain regular headways, and keep a consistent travel time. Others like Strathman (2003) describe service reliability as the ability to adhere to a schedule and keep schedule-related delays, run-time delays, run-time variations, headway delays, and headway variations to a minimum. In other words, transit services are reliable when the buses run on time and adhere to the schedule, maintain regular headways, and ensure a uniform travel time.

In order to analyze the performance of services, it is imperative to have a service standard. Although traditional measures are similar to the transit agency point of view, there is a new perspective on analyzing performance from the passenger point of view. Traditionally, transit agencies are more concerned with schedule adherence, regular headway, capacity, passenger load, etc., whereas passengers are more concerned with waiting time, availability of seats, comfort, etc.

TCRP Report 165 mentions different measures related to service reliability, such as on-time performance (OTP), coefficient of headway variations, missed trips, and distance traveled between mechanical breakdowns. However, these measures cannot solely explain unreliable service, as measures such as OTP alone cannot distinguish between early and late departures, and coefficient headway variations fail to explain whether bus bunching or gaps are causing the variations (Figliozi, 2012).

According to TCRP Report 165, transit agencies use OTP as reliability measures for headways greater than 10 minutes, and coefficient headway variations for headways less than 10 minutes. In addition, some agencies also use information such as missed trips, headway adherence, schedule efficiency, passenger load ratios, etc., to better document the performance of services.

On-time Performance (OTP)

On-time performance is a percentage value used to indicate whether buses arrive or depart late, on time, or early. TCRP Report 165 proposes 0 to 5 minutes as the threshold values because most passengers adjust their time according to scheduled arrival time, and no earlier than that. Table 1 in TCRP Report 165 presents the Transit Level of Service (LOS) values.

Table 1: Transit Level of Service

LOS	On - Time Percentages
A	95-100%
B	90-94.9%
C	85-89.9%
D	80-84.9%
E	75-79.9%
F	<75%

Note that some agencies use different measures of OTP. For example, the Chicago Transit Agency (CTA), New York City Transit (NYCT), and Tri-Met (Portland) use 1 minute early to 5 minutes late, and Miami-Dade Transit and other agencies use 2 minutes early to 5 minutes late (Cevallos, 2010).

2.1.2 Headway Regularity

The coefficient of headway variation, as shown in Table 2, is another commonly used indicator to gauge the services, and is used to measure the deviation of its headways from the mean, as follows:

$$C_{vh} = \frac{\text{standard deviation of headway deviations}}{\text{mean scheduled headway}}$$

where, C_{vh} = coefficient of variation of headways

Table 2: Headway Coefficient of Variation

LOS	C _{vh}	P (h _j > 0.5 h)
A	0-0.21	≤ 1%
B	0.22-0.30	≤ 10%
C	0.31-0.39	≤ 20%
D	0.40-0.52	≤ 33%
E	0.53-0.74	≤ 50%
F	≥ 0.75	≥ 50%

(Source: TCRP Report 100)

Other attributes used by transit agencies include missed trips, passenger load factor ratios, etc., for performance measurement (Furth, 2003).

Many research studies use variables relating to waiting time for passengers, even though it is not common in practice. Previous studies (Welding 1957, Osuna 1972, Heap 1976) developed methods to measure waiting time in terms of headway and random passenger arrival, as shown below:

$$E (T_{i,j}) = E (H_{i,j}) / 2 * (1 + CoV^2 (H_{i,j})) ,$$

Where,

$E (T_{i,j})$ = expected passenger waiting time at stop j on line i

$E (H_{i,j})$ = expected headway at line I at stop j

In the case of the planned arrival of passengers, the average waiting time is half of the headway, as given below:

$$E (T_{i,j}) = \frac{E(T_{i,j})}{2} * (CoV^2 (H_{i,j}))$$

Few studies (Furth 2006, Oort 2011) consider additional buffer time or the 95th percentile travel time. Other studies (Furth 2006, Casello 2009) quantify user costs associated with reliability.

Limitation of Traditional Performance Indicators

One of the major weaknesses of these indicators is that they are measured at the aggregated level. For example, the coefficient of variations cannot separate whether a gap or bunching is occurring. To overcome these limitations, several studies propose alternative indicators for transit performance, such as weighted delay index (Camus, 2005), punctuality index based on routes, deviation index based on stops, evenness index based in stops (Chen, 2009), earliness index, width index, and second order stochastic dominance (Sabeti, 2013). Other studies (Lin, 2008) propose a combined standard to measure the service reliability.

Possible Causes of Service Unreliability

Service reliability is affected by both external and internal elements, such as traffic signals, traffic conditions, unexpected situations, route structure, stop patterns, scheduling and fare collection practices. A comprehensive review on studies identifying causes of service unreliability is found in research conducted by the following authors: Abkowitz (1983), Turnquist (1981), Levinson (1991), and Ceder (2007). Recent studies (Cham, 2006, Van Oort, 2011) attribute schedule deviations at terminals and variability of trip running time as the main causes of service unreliability (see Table 3).

Table 3: Possible Causes of Service Variability in Public Transport

Internal Causes	Terminal Time Variability Trip Time Variability (Driving time , Dwell time, Stopping time)	External Causes
Infrastructure design		Weather
Service Network design		Driver, Traveler Behavior
Driver Behavior		
Vehicle Availability		Irregular Passenger loads
Crew Availability		Other Traffic
Other Public Transport		

Cham (2006) proposes a framework for identifying the reason for transit service unreliability and offers strategies to solve these problems. Figure 1 shows a modified version of Cham's work.

Figure 1: Transit Unreliability Framework

Other research studies (Mandelzys 2010, Saberi, 2012, Peng, 2008), as depicted in Table 5, analyze detailed strategy levels in order to find reasons at the stop level for the late arrival or early departure of buses.

Various papers present different ways to identify causes of service unreliability. Studies such as Hammerle (2005), Golani (2006), and Feng (2011), demonstrate more advanced graphical methods for identifying unreliable service problems at different stops and time periods using time-space diagrams or the Geographic Information System (GIS). Figure 2 is a graphic depiction of bunching issues.

Figure 2: Identification of Bunching using a Time-Space Diagram

Several studies use data mining, while others use advanced statistical analysis for identifying bus bunching, which is summarized below:

Moreira-Matias (2012) proposed a methodology of identifying bus bunching using a sequence-mining algorithm. The author used AVL historical data for every bus pair to develop headway sequences, and then separate those that cause bus bunching. The results show that headway deviation may propagate from one point to another. One finding suggests adjusting the timetable or adding more slack times to vulnerable points so that service is easily restored.

Cevallos (2007) provided an overview of the potential use of data mining for improving transit service operation. The paper also describes the methodologies used to set up data management, data mining, reporting services, and performance measures.

El-Geneidy (2007) discussed the methodology used to investigate the performance and reliability problems along any problematic route using archived AVL and Automated Passenger Counter (APC) data. El-Geneidy analyzed performance at two levels: trip level and timepoint segment level. At the trip level, analyses were performed to assess the running time or scheduling problems. First, layover or recovery is derived from the difference between the 95th percentile running time and the selected benchmark (mean or median running time), and then the value is compared with the actual average layover time to determine whether any rescheduling is required. For the timepoint segment level analysis, data are first segmented between two consecutive points, and statistical analyses are performed to understand the factors that cause the reliability and

performance decline along this route. Regression models are developed for the running time, running time deviation, headway deviation, running time variations to examine the impact of multiple route characteristics (such as number of physical stops, number of actual stops, boarding, alighting, lift usage, driver's experience, schedule delay at start, headway delay at start, passenger load, segment location along the pattern, segment length, etc.) on performance and reliability. The unique feature of this approach is that it allows transit planners to identify the impact of specific characteristics on a route's overall performance.

Research by Peng (2009) focused on two important issues of transit reliability: bus bunching and service gaps use AVL data together with schedule data. This paper considers bus bunching as the actual headway of 1 minute or less, and a large service gap as a headway ratio of 1.5 or larger. First, service gaps and bus bunching are identified by time period and the use of sample runs. This paper also attempts to find patterns. Second, in order to test whether service gaps or bunching propagates into the next timepoints along the route, regression models are developed by considering a dependent variable as the actual headway at a given timepoint and an independent variable as the actual headway at previous timepoints. This study also performed analyses to identify the type of conditions under which service gaps tend to occur so that a warning system is developed to alert operators. To accomplish this step, six probable conditions for bus bunching or service gaps based on actual headway difference than the schedule are set, and later, statistical analyses are performed to find the threshold values for every condition. For example, this report found that when the first bus is 1.5 minutes quicker than the scheduled travel time, and the second bus is 4-5 minutes slower than the scheduled travel time, there is a 95% probability that a large gap will occur during the period of 6:00 a.m. to 9:00 a.m. for the third condition. Different threshold values can be determined for other probability levels, such as 50%, 75% and 95%, which will alert the operator before the event occurs.

Mandelzys (2010) proposed a methodology to analyze schedule adherence and determine the causes of poor schedule adherence by using AVL/APC data. While analyzing the performance, the author considers three variables: Not "on-time" percentages (traditional measure) and not "on-time" percentages from the passenger perspective, taking into account the percentage of passengers affected while onboard, or waiting at stops. This paper also proposed a flowchart that identifies the causes of late arrivals and early departures, which is useful for small- or medium-size transit agencies. Table 4 shows the possible causes for early and late arrivals. The end portion of this paper shows the statistics of causes for late arrival and early arrival at stops, and proposes strategies to minimize these causes.

Table 4: Possible Causes for Late Arrivals and Early Departures

Problem	Possible Causes	Potential "Real-World" Reasons
Late Arrival	Travel time from previous stop took longer than scheduled and $(I_{i,j} = 1, (A_{a_{i,j}} - A_{a_{i-1,j}}) > (S_{a_{i,j}} - S_{a_{i-1,j}}))$	<ul style="list-style-type: none"> • Traffic reasons (congestion, inclement weather, signal timing, etc.) • High demand for intermediate (non-timepoint) stops • Unscheduled stops • Etc.
	Dwell time at previous stop was longer than scheduled and $(I_{i,j} = 1, (A_{d_{i-1,j}} - A_{a_{i-1,j}}) > \max((S_{d_{i-1,j}} - S_{a_{i-1,j}}), D_t))$ ³	<ul style="list-style-type: none"> • High passenger activity (on/off) • Difficulty rejoining traffic stream • Lift/ramp use • Etc.
	Arrived at previous stop late and $(I_{i,j} = 1, (A_{a_{i-1,j}}) > (S_{a_{i-1,j}}))$	<ul style="list-style-type: none"> • Upstream causes
Early Departure	Dwell time at current stop was less than scheduled and $(E_{i,j} = 1, (A_{d_{i,j}} - A_{a_{i,j}}) < (S_{d_{i,j}} - S_{a_{i,j}}))$	<ul style="list-style-type: none"> • Low passenger activity (on/off)
	Travel time from previous stop was lower than scheduled and $(E_{i,j} = 1, (A_{a_{i,j}} - A_{d_{i-1,j}}) < (S_{a_{i,j}} - S_{d_{i-1,j}}))$	<ul style="list-style-type: none"> • Traffic reasons (lower than expected congestion, etc.) • Low demand at intermediate (non-timepoint stops) • Etc.
	Departed previous stop early and $(E_{i,j} = 1, (A_{d_{i-1,j}}) < (S_{d_{i-1,j}}))$	<ul style="list-style-type: none"> • Upstream causes

Restoration Strategies

Previous studies propose and describe different ways to restore services. As proposed by Turnquist (1980), restoration strategies are categorized as planning and real-time strategies. However, there are big differences between these strategies in terms of the resources, time spent and time required to obtain the results. Usually, strategies related to planning involve time, money and results, which are realized in the long run; whereas real-time strategies can immediately produce benefits, but only to a certain extent.

Van Oort (2011) proposes different strategies, as presented in Table 5, at both the tactical (scheduling, holding) and strategic design levels (network design, terminal design, etc.), and assesses their impacts on service reliability.

Table 5: Strategies on Service Reliability

Level	Strategies	
Strategic	Infrastructure Network:	Terminal design Stop design Exclusive lanes Traffic Signal priority
	Service Network:	Line coordination, Line length, Line synchronization, Stopping Distance
Tactical		Trip Time determination, Vehicle Holding

More importantly, the author proposes a new concept called the “feed forward” loop from the operational point of view to predict the disturbances at both the strategic and tactical levels beforehand, as well as to incorporate this information into the plan. Figure 3 presents a diagram with the proposed planning tools (Van Oort, 2011).

Figure 3: Proposed Planning Tools

There are many studies on improving service scheduling. While developing a schedule, the main concern is not the addition of more slack time. Inserting more slack time in the schedule’s travel time reduces the overall average operating speed, thus increasing the travel time and more importantly, operational costs (Furth, 2006). Usually, slack times are put either into timepoints/holding points, or at stations as recovery time. With the availability of AVL data, it is now easier to find these values. Strathman (2002) highlights the comparison of benefits for using different threshold values in schedule determination. Figure 4 presents a graph that helps visualize bus running times and layover times (Kimpel, 2004).

- 1) The difference between the 95th percentile run-time and the median, or mean.

- 2) The minimum amount specified under the labor contract (10% of the median run-time).
- 3) Any amount used by schedulers as a rule of thumb (i.e., 18% of the median run-time).

Figure 4: Determination of Optimal Bus Running and Layover Times

Another branch of studies (Pangilinan, 2008; Daganzo, 2009; Bartholdi, 2012; Chen, 2013) looked into different holding strategies, which are paramount for alleviating bus bunching. Daganzo (2009) proposed targeted headway base measures to compare bus headway with predefined headway. If the value is smaller than the predefined headway, then the buses are getting closer in distance to each other. As a result, the following bus is held until it returns to the scheduled headway.

Bartholdi (2012) proposes a more practical, effective technique. In this technique, buses are held at a control point for a definite amount of time (either a fraction of headway or definite amount of time from the previous departed bus, whichever is smaller). The Georgia Institute of Technology successfully tested the theory.

Pangilinan (2008) evaluated an innovative control strategy called the Prefol strategy, initially presented by Turnquist (1982), which significantly reduces the headway variations (coefficient of headway variations). In the Prefol strategy, buses are held until the headways between the

preceding or following bus and the bus are equal, thus preventing the formation of large gaps. The author also highlights one important constraint: that the CTA system is missing the simultaneous observation of all timepoints, which is necessary for the assessment of all of the benefits of AVL.

Chen (2013) proposed a methodology for holding a bus at multiple control points. The authors developed mathematical models, and showed that holding a bus at multiple points minimizes the bus waiting time and improves the headway performance.

Other studies (i.e., Cevallos, 2010; Fattouche, 2011) focused on improving the scheduling of service reliability. Cevallos (2010) proposed an approach to improve service reliability by adjusting the schedule timetable. The author first performs different statistical analysis on the AVL data of Miami-Dade Transit to determine the distribution of schedule adherence, and then modifies the scheduled time for trips in order to maximize the number of trips that fall within the specified “on-time” definition. In addition, another method of adjusting a schedule is discussed, which continues until the optimal values are reached.

Fattouche (2007) proposed a cost-based scheduling procedure for improving service reliability. The main purpose of Fattouche’s research work was to develop a model that minimized the costs associated with waiting passengers and onboard passengers, taking into consideration that increasing the running time meant increasing the operation costs for the transit agencies. The model uses AVL and APC data to determine segment running time distribution, average load, and passenger arrival rate, and then calculates the headway coefficient of variation, mean segment running time, and mean time spent on a vehicle. Then, these values are converted into monetary values, which are used to develop a generalized cost model function. The findings of this research show the development of an optimized schedule from both customer and transit agency point of views.

Following is a summary of operation strategies practiced by several transit agencies, which is helpful in understanding the extent to which theoretical knowledge is used in the real world. Table 6 presents different operational strategies, and it is based on the work by Peng in 2008.

Table 6: Agencies' Operational Strategies

Transit Agency	Real-time Operation
King County Metro	Inserting a bus, skipping stops.
	Expressing.
	Short-turning, queue jumping, traffic signal priority.
Los Angeles	Equipped with AVL in 99% of buses. Mainly relays, but no holding.
MBTA	Expressing, short-turning, no stoppage at terminal, holding at terminals, at mid-ways.
Metro St. Louis	AVLs are mostly used for adjusting running time. Strategic buses are to carry on the overload or address a missed trip. Strategies like deadhead, no stopping at terminals, etc., are used.
New York City Transit (NYCT)	Holding at control points. Because of a complex network, other measures like short-turning and expressing are difficult to implement.
Tri-County Metropolitan Transportation District of Oregon (Tri-Met)	Short-turning, expressing, traffic signal priority, possibility of using a headway-based holding. Regular reports on missing trips and lost service hours are published to regulate performance.
VIA Metropolitan Transit	Uses the "TransitMaster BusOP" application, which provided complete solutions for service restoration. Strategies include inserting buses, turn-backs, bus changes, schedule changes, and short loops.
Washington Metropolitan Area Transit Authority (WMATA)	Uses real-time fleet management and voice radio communication to restore services. Restoring strategies include inserting buses and reblocking. Efforts like awarding or punishing the operator, traffic signal priority, access to camera (ITS), etc., are implemented.

Literature Review Summary

According to the literature review, this study finds that there are several weaknesses in indicators commonly used for the performance measurement of transit services. For example, “on-time performance” or “coefficient of headway deviations” cannot distinguish between the earliness and lateness of services, or between gap and bunching. One of the most commonly used indicators in research, but absent in the real world, is passenger waiting time.

Transit services can degrade for many reasons, both externally and internally. For instance, the cause may vary from inadequate service network design, mismanagement of resources (vehicle, crew), and very high passenger load, to more stochastic causes, such as weather, driver behavior, policy, etc. However, previous works point out the “deviation in terminal” and “variability of trip running time” as root causes for service degradation. The most important implication of these

factors is that few measures are needed during the planning stage. For example, if there is any error in terminal design, the services are more likely to be hampered on a regular basis, and any step to correct this problem later will be costly, time-consuming, and inconvenient to passengers.

Several methodologies are derived from the proposed indicators in the TCRP Report 165. Most of the studies recommended first developing a Time-Space diagram using historical Automatic Vehicle Location (AVL) data for different routes at different time periods (days, time of day), and then identified stops at different time periods where bus bunching began to occur. Subsequently, few researchers developed advanced statistical models (regression models) to understand the influence of service, network characteristics (number of stops, distance among stops, dwell time, passenger load, etc.) upon service degradation. Others proposed methods to identify the most vulnerable points along the service routes using advanced methods like data mining, and took the correct steps to solve the probable points.

However, only few studies aim to find the pattern of service failure. Theoretically, transit services usually do not suddenly fail. When a bus is late at any station, it is more likely to be late at consecutive stops, as it has to board more passengers than before. The buses will bunch if no remedies are employed, as the following bus is supposed to take fewer passengers. If this phenomenon continues, the service will likely fail at one stage. One relevant study (Peng, 2009) investigates the pattern of service failure in terms of gap and bunching occurrences. Although the findings of the study are promising, the threshold values proposed by the study are only applicable for the study area. Therefore, there is a need for more detailed studies to find the possible patterns of failure and a generalized framework that can be applied to any area.

Most strategies to restore the services proposed by different studies are categorized by two types: Planning stage strategy and operation stage strategy. While planning stage strategies require more resources (time, money, etc.), and results are realized in longer time spans, the results of operation time strategies are immediate and less costly. However, the most effective strategies should include data from both types.

The most commonly used planning strategy is to improve service scheduling. One method involves the insertion of slack times at timepoints, and many threshold values are suggested for this method. One critical issue, however, is that it is a tradeoff between passenger waiting time and transit agency policy. Another method proposed by researchers is the adjustment of the schedule timetable or the determination of the optimized schedule by considering both passenger convenience and transit operation cost to improve service reliability. Other studies focus on planning and real-time operation strategies, as presented in Table 7.

Table 7: Planning and Real-Time Operational Strategies

Phase	Strategy	Stage of Implementation
Planning Stage	Infrastructure Design (Terminals, Stops, Short-turn facilities), Network Design, Traffic Signal Preemption, Queue Jumper	–Pre-Operation
	Running Time	Operation
Real-time Operation	Holing, Expressing, Deadheading, Skipping stops, short-turning, adding vehicles	Operation

Lastly, the Automatic Vehicle Location (AVL) system plays an important role in overcoming most of the limitations discussed above, whether related to service performance indicators or service improvements or restoration. Although most agencies use the AVL system for service monitoring, only few agencies make the most use of immense amounts of data. Nonetheless, the systems currently used for service monitoring or bus dispatching are limited in the area of early indications of possible service failure. Most of the time, the problems are detected after they have already occurred. Hence, this study attempts to help solve these types of problems by presenting ideas and examples on how to use AVL data to improve service reliability.

ASSESSMENT OF CAD/AVL SYSTEMS

Fleet management and other intelligent vehicle location and dispatch applications have traditionally sought to organize transit system data into a standard, easily navigable format. Centralized and mobile CAD/AVL user interfaces provide information to dispatchers, such as a view of asset locations, communication through a call and incident management queue, and vehicle on-time performance. The model of this arrangement relies on well-trained and resourced dispatching staff to interpret this array of data and the ability to make decisions that maintain acceptable levels of transit system performance.

The process flow for translating incoming information into decision-making is becoming more automated. One method of the automation of responses to certain predefined service conditions is the development of service degradation identification methods. As the quality of service begins to drift from user set thresholds, as defined by the number of concurrently running deviations from a scheduled and fixed route, restoration plans are put into action. This section reviews the current state of practice with regard to the availability of these tools, as well as where these tools could potentially reside within the current suite and capability of commercially available fleet management systems.

Innovative Use of AVL Data

Currently, many transit agencies use an automated fleet management system to restore services in real time. In one related study, Strathman (2003) discusses how the use of the automated bus dispatching system (BDS) by the Tri-County Metropolitan Transportation District of Oregon (Tri-Met) has yielded benefits for both the agency and public.

Hammerle (2005) discussed the methods used to evaluate bus operation through the use of AVL and APC data. Hammerle first examined schedule adherence, service reliability, and headway regularity separately by using stop level AVL and APC data, and later examined how these factors all tied together. For schedule adherence, earliness or lateness at stops along the route is measured and plotted in graphs during the morning peak hours. For service reliability, a time-space curve (arrival time versus the stop name) is drawn using the data, which is used to investigate bunching problems in the headway irregularity location. While measuring the headway regularity, two cases are considered: an extreme condition (headway less than 30 seconds) and a less extreme condition (the headway is more than a half-scheduled headway). The research also included measuring the service gap using the stop level data, which considered the fact that the actual headway was twice greater than the occurrence of the scheduled headway gap. Lastly, by examining the combined effects of all factors (schedule adherence, service reliability, headway regularity), this paper found that CTA buses departed early at several stations, which created bunching with the previous trip and service gaps with the following trip. However, this can be easily resolved by correcting early terminal departure.

Cham (2006) first proposed a practical framework to assess bus service reliability using AVL/APC and analyzed the service reliability of a specific route (Massachusetts Bay Transportation's Silver Line Washington Street). The proposed framework consists of mainly three parts:

1. Characterizing the service reliability through service measures and performance reports. This part also describes key data inputs used to measure the performance.
2. Identifying the causes of reliability problems.
3. Developing a section of strategies based on the previous process (Step 2) to improve reliability.

The purpose for characterizing service reliability is to give transit providers a complete guideline to assess the overall route service condition. This includes determination of the type of data input needed for performance analysis and service attributes needed for calculation, defining service measures and their thresholds to investigate overall performance, and proposing a format of the report for presentation. After applying the proposed methods to the Massachusetts Bay Transportation's Silver Line Washington Street route, the author found that there were significant variations in service attributes, even though service attributes like mean running time and mean headway were too close to schedule. The main cause of this service unreliability could be poor supervision on the terminals and operator behavior. The report also states that more than 50% of deviations at points other than terminals begin at the deviations at the terminals. However, due to an absence of APC data, the abnormal passenger loads are not directly measured. Lastly, based on the reliability problems found after the analysis, preventative strategies like operator training, corrective strategies and signal priorities are proposed in this report.

Golani's research (2006) performed the following tasks: 1) Reviewed the current use of AVL data from the Champaign-Urbana Mass Transit District (CUMTD) for transit planning and operation; 2) Presented the various methods of visualizing data to investigate the causes of service degradation with the use of GIS; and 3) Suggested a novel use of archived Computer-Aided Dispatch (CAD) message data to investigate the response performance. The current use of AVL and APC data by the CUMTD for planning and operation, which are mentioned in the paper, are shown in Table 8.

Table 8: CUMTD Uses of AVL and APC Data

Data from AVL / APC	Unit	Application in Planning & Opt.
Boarding and loads by blocks	Minimum, maximum & average boarding	Allocation of appropriate bus sizes to the blocks
Boarding for all stops	Monthly & daily boarding	Planning of infrastructure and passenger amenities at stations
Running times	All stops	Reviewing of Scheduling
Ridership per trip	All trips for a route	Possibility of trip Consolidation or Express service
Passengers per revenue hour		Indicator of Overall route performance
Percent of on-time stops	-2 (early) to 5 minutes (late) used as benchmark	Indicator of Overall route performance

This paper also shows how GIS is used to visualize schedule deviation and passenger count data along the route in order to identify patterns in schedule deviation and ridership along the problem blocks, trips, stops and times. Lastly, it proposes a novel use of archived screen messages of CAD systems to measure the efficiency of the dispatch systems by analyzing the response times of dispatchers in the control room when a request to communicate is initiated by an operator.

Wong (2010) discussed the AVL systems of the iBus in London and showed how these data are used for better transit operations. The real-time information received at the AVL center in regular intervals is used for three purposes: bus fleet management, real-time passenger information, and bus priority at traffic signals. After processing the data, real-time passenger information is disseminated through the roadside beacons located along the bus route via text messaging and the internet. With the provision of locating every bus against tabular representation in geographical maps, the performance of each bus is monitored in real time, and fleets are managed in the network accordingly. Unlike conventional infrastructure-based detection systems, the iBus system uses predefined virtual detector coordinates set in the onboard unit of the bus to trigger a priority request by transmitting a bus priority telegram to the traffic signal controller. The unique feature of this request is that the bus initiates different priorities based on the earliness/lateness of the bus relative to the predefined timetable to reduce the impact on other traffic, and leads to more regular and improved journey times for buses overall. The applications for post-processing data management are improved service monitoring, bus rescheduling, and dwell time estimation.

Tétreault (2009) used AVL and APC data to design the new limited stop service for bus routes (67 Saint-Michel, a high frequency route in the STM system) in Montréal, Québec, Canada. The author developed multivariate linear regression models of running times for four possible cases (only transfer stops, or stops having upper quartile passenger activities, selected stops, or stops with definite spacing). In the development process of the models, different variables related to stops, passenger activity (boarding and alighting), bus floor type, direction, time of day, day of week, weather, delays from the beginning of trip, etc., were considered. Since the author is interested in

comparing total journey time, the time required to get to the stations is estimated using a street network. The research studies these runtimes under different conditions (pessimistic, optimistic, realistic), as it is not possible to figure out the actual number of passengers that will shift from regular to limited bus services. The results find that services with definite stop spacing, along with higher passenger activity, are the most efficient.

Carrasco (2012) evaluated the transit service performance in Zurich, Switzerland on route and stop level using AVL data. The study analyzed different performance measures, such as schedule deviation, headway regularity, travel time, vehicle speed profile, and passenger waiting time during different time periods of the day. The findings show that AVL data can be very useful in finding stops and periods of time when services are most unreliable.

Transit Operations Decision Support Systems (TODSS)

The Federal Transit Administration (FTA) initiated a prototype next generation CAD/AVL system conceived as a Transit Operations Decision Support System (TODSS) (Mitretek Systems 2004, FTA 2010). TODSS was developed in response to feedback from transit agencies, which found that existing CAD/AVL systems were good at providing a large quantity of data but lacked concise, actionable information to proactively manage transit fleets in revenue service. TODSS specifically sought to:

- Develop a common understanding between agency needs and product vendor development.
- Minimize vendor customization costs for future CAD/AVL procurements.
- Provide agencies with assistance to develop procurement specifications for a TODSS.

At its core, TODSS helped translate the needs of transit operators to the experts in the private sector to develop a CAD/AVL that could deploy when fixed-route service deviates from the expected performance and supply a dispatcher with a restoration strategy based on constraining factors. At a minimum, these decision support recommendations could include adjustment of vehicle headways, dispatching replacement or additional vehicles, or reporting incidences.

These adjustments could only occur when the sources of information are continually monitored and only those events requiring dispatcher attention are displayed, along with corresponding service restoration action plans. Multiple sources of information are required to make these assessments and corresponding rules to trigger the plans. Measures of demand for service, including peak loads, route loads, and load over a segment of the route and traffic conditions are considered. Characteristics of the route, including turn-around points, detour routes, scheduled deadheads, route branches, common trunks, and length of trip were also included. The operating environment, including the garage location, relief points, headway intervals, and vehicle and operator availability are also involved.

These systems would also require notification of service interruptions by external non-automatic data sources in order for TODSS to provide a complete set of source information to identify incidents. The sources within the demonstration site environment include customer cell phone calls, contact from other control centers, or calls from local road commissions.

Computer-Aided Dispatch Systems with Transit Quality of Service Alerts

The first generation of CAD/AVL software provided standard alert messages specific to a single vehicle and its deviation from expected performance. This includes vehicles ahead or behind schedule by a user-defined setting. This could also include an off-route condition where a vehicle deviates a certain distance from the route it is traveling. There is also the capability to identify vehicles at or near maximum capacity, as well as those with message identifiers sent to a variety of vehicle health conditions.

These conditions are represented on a map or in the CAD window with different colors that identify which service rule the vehicle was out of compliance with. The vehicle queue is then reordered to place these buses and routes in the most visible location to attract the dispatcher's attention. A CAD/AVL may also have a separate window or queue for buses in a degraded service condition. An audible tone is associated with the condition, calling for human intervention. Quite often, these conditions occur independently, without any context to the impacts made to the entire transit network, such as the potential for missed transfers and vehicle bunching. Dispatch staff must interpret these competing signals and decide what action, if any, should be taken to restore the service.

Methods to more dynamically represent the collection of multiple degraded service conditions have evolved incrementally. The use of what one vendor terms the "rubber band view" provides a visual representation of the impacts of bus bunching in a headway-based transit network. In an optimum service environment, the rubber band view should show an evenly distributed series of buses, each maintaining the correct headway. As vehicles begin to draw closer to each other along the rubber band, the correct headway spacing between vehicles is no longer maintained, and the service becomes degraded. While this view provides a grouping of multiple assets with a failing quality of service, the dispatcher must still review and interpret this information and take corrective actions to restore service.

TODSS Architecture and Concept of Operations

An operational process and procedure review is necessary for developing the local requirements for each operational scenario. The operational scenarios can be broken down into four parts, as shown in Figure 5 (FTA, 2010), to assist with the development of business rules to manage incidents. The activities include defining the following:

- **Initial Actions** – Actions that describe which dispatcher tasks need to be completed after receiving a service disruption notification.
- **Service Restoration Strategy** – Includes dispatcher response options based on the results of the initial actions.
- **Follow-up Actions** – Actions that the dispatcher should perform after exercising restoration options.
- **Dispatch Document References** – References that contain supporting material for further information related to the operational scenario and are readily accessible to the dispatcher.

The first process requiring action is a definition of rules to interpret incoming data messages external to the dispatch center. This includes setting priority levels for dispatchers to respond consistently to events processed as service incidents.

Service restoration strategies are based on the type of incident notification and operational scenario and are provided to dispatchers for a uniform agency approach to service management. The full set of service restoration strategies, as identified by the TODSS Working Group, include:

- Vehicle jumping with an available vehicle (parked, staged, pulled-in) to replace a vehicle that became unavailable (breakdown, delayed)
- Shift the schedule time frame
- Eliminate a departure
- Insert a departure
- Modify schedule running times
- Wait at a bus stop or transfer point
- Bus changes
- Pass on the route
- Exchange drivers
- Route deviations
- Short-turns
- Relay vehicles

Figure 5: Concept of Operations TODSS System Design

The initial concept of operations includes the following preliminary operational scenarios:

- Vehicle Breakdown
- Train Crossing and Open Bridge
- Late Startup and Pullout
- Bus Bridge
- System Failure
- Evacuations
- Passenger Facilities Emergencies
- Emergency Phone Line
- Contractor Accident Report
- Stranded Paratransit Customer

In a 2012 study, the benefits of the demonstration site that implemented TODSS were documented, which included improved dispatcher service performance, improved quality of dispatcher

responses, greater uniformity of action among dispatchers, and an increase in real-time communication throughout its operations from the dispatch center to agency-wide (Vanchugov, 2012).

The TODSS CAD/AVL framework is rules-based, which requires input from a variety of data sources. Hence, it appears it would not be difficult to allow a predictive service degradation methodology to become another trigger for a series of service restoration strategies.

Nearly a dozen transit agencies, including those in Chicago, Dallas, Cincinnati, Des Moines, San Antonio and other cities, are in various stages of CAD/AVL upgrades that will include some level of TODSS or Intelligent Decision Support (IDS) capability. As agencies become more accustomed to a higher level of performance within their fleet management systems, predictors of service degradation will be an expected function of these suites of tools.

STUDY APPROACH

In this research, transit service reliability is addressed from the transit agency point of view and focuses on improving on-time performance and monitoring schedule adherence and bus bunching to prevent service degradation. It also makes use of static and real-time data from AVL systems. Data from the HART General Transit Feed Specification (GTFS) and the MDT AVL system are used as examples of how AVL data and derived information can be instrumental for improving service reliability. The HART datasets include both static and real-time GTFS data. The MDT datasets include archived data at the timepoint level and data compiled from real-time outputs.

In this section, the emphasis is mainly on GTFS, as the information on AVL systems is widely available in the literature. Nevertheless, the Case Studies section presents examples of using both datasets. The GTFS dataset is populated with data from AVL systems. It was included in this study with the hope that it will create opportunities for transit data standardization, provide useful information about GTFS, and encourage other researchers to use this readily available data for future studies.

For the on-time performance (OTP) portion, the following items are considered: data quality, schedule adherence parameters, missing records, extreme values or outliers, end-of-line (EOL), and other issues that may have an impact on the way OTP is calculated, measured, monitored, and managed. Service degradation is addressed through the monitoring of bus schedule adherence and conditions such as the fluctuation of spacing among vehicles that can lead to bus bunching.

General Transit Feed Specification (GTFS)

The General Transit Feed Specification (GTFS) serves as a common platform for public transportation schedules and associated geographic information. The GTFS was developed by Tri-Met in 2005 in Portland, Oregon; Tri-Met began working with Google on incorporating transit agency data into their trip planners. Initially, it was named Google Transit Feed Specification and was easily maintainable, scalable, and could be incorporated into Google Maps. However, later it was renamed “General Transit Feed Specification,” and Google began offering trip planning services at no cost to agencies that provided their transit data in the GTFS format. Presently, GTFS has become one of the most popular data formats in the world, with an increasing number of agencies choosing to share their transit data with the public (TransitWiki. General Transit Feed Specification).

A GTFS static feed is composed of a series of text files collected in a ZIP file. Each file manifests a particular aspect of transit information: stops, routes, trips, and other schedule data. The details of each file are defined in the GTFS reference, as shown in Appendix A. A transit agency can produce a GTFS feed to share the information with developers, who can write applications to provide transit information to the public. GTFS can be used to develop trip planners, timetable publishers, and many other applications (Google Developers. Static Transit).

There are many approaches to creating a feed in a variety of formats. For instance, there is widespread use of Excel spreadsheet tools. These tools are usually open source, such as the XLS Tools for Google Transit Feed Specification developed by Robert Heitzman (<https://sites.google.com/site/rheitzman/>), so that they can be used by anyone.

Creating a GTFS Static Feed

The following are steps to create a GTFS feed (Google Developers. Static Transit: <https://developers.google.com/transit/gtfs/>).

1. Examine the GTFS examples.
2. Create your own feed using the GTFS Reference as a guide.
3. Test your feed using validation tools. There are two open source tools available for testing feeds in the GTFS format, as follows:
 - **Feedvalidator**: This tool is used to verify that the feed data file matches the specification defined in this document. More information can be found at <https://github.com/google/transitfeed/wiki/FeedValidator>.
 - **Schedule Viewer**: This application is used to view the feed data represented on a map. This is not representative of how the data will look in other applications; it is a basic tool for testing. Users should examine routes and schedules to ensure that the data feed correctly represents the transit agency's system. More information can be found at <https://developers.google.com/transit/tools#GtfsValidation>.
4. Publish the feed.

Make a Transit Feed Publicly Available

Many applications are compatible with data in the GTFS format. The simplest way to make a feed public is to host it on a web server and publish an announcement that makes it available for use. Several ways in which the interested software developers learn about public feeds include:

A list of transit agencies that provide public feeds, which is available on the GoogleTransitDataFeed project site. Examples include Tri-Met, BART, Dart, HART, etc. More information can be found at <https://code.google.com/p/googletransitdatafeed/wiki/PublicFeeds>. The GTFS Data Exchange website allows developers to subscribe to announcements about new and updated feeds. Additional information can be found at <http://www.gtfs-data-exchange.com/>.

Submit a Transit Feed to Google

Public agencies that oversee public transportation for cities can use the GTFS specification to provide schedules and geographic information to Google Maps and other Google applications that show transit information. Figure 6 is an image used by Google that depicts a transit trip in San Diego. Following are the steps that explain how to add transit data to Google Maps.

Figure 6: Transit Data in Google Maps

(Source: <http://maps.google.com/help/maps/mapcontent/transit/participate.html>)

Steps for Adding Transit Feed to Google Maps

1. Prepare a data feed according to the General Transit Feed Specification Reference.
2. Validate the feed using the Feed Validator.
3. Inspect the feed in the Schedule Viewer.
4. Zip the files in your feed. Name the zip file **google_transit.zip**.
5. Host the feed on a web server for Google to retrieve it (it supports both HTTP and HTTPS).
6. Contact the Google Transit team to sign up for the partnership.
7. Google will communicate with the user to set up a private preview and have the agency complete an online agreement before launch.
8. The agency will test the data in the private preview until the result is satisfactory.
9. Launch!

Host a Feed

The following steps demonstrate how to host a GTFS feed:

1. Place the feed in a directory that will always keep the same name, for example, http://myserver.agency.com/current/google_transit.zip.
2. Enable the directory listing in the directory where the feed file will be hosted. The feed should be the only file in the directory.
3. To change or update the feed, replace the old google_transit.zip with a new google_transit.zip. The new data will completely overwrite the old data. As a result, please ensure that your new feed has complete data as of the posting date. For example, if you post a new dataset on 12-8, the data must have service as of 12-8.

4. The IT/Networking teams should know that Google Maps periodically fetches transit feed data from the location that you specify, so that they do not change file permissions for your feed and otherwise block or break the data fetching process.

Validate the Data Feed:

Google has created a few open source tools to help agencies check the feed quality.

- **Feed Validator:** This tool checks the feed format. Feed should not be submitted to Google until it is free of format errors.
- **Schedule Viewer:** This tool helps agencies visualize every route and stops in Google Maps. Agencies use this tool to systematically check every route for stop locations, stop sequence, vehicle speed, and other important issues.

Once Google accepts the transit feed, Google will build a private preview for the agency to test. Since the agency knows the area the best, it is important that the agency thoroughly test various types of routes, such as the most popular routes and routes used on holidays or weekends. Google will not launch the data until the agency confirms its quality, signs an online agreement, and confirms that the feed has passed Google engineer's inspection.

Feed File Requirements

A General Transit Feed Specification is provided in this paper as Appendix A (Google Developers. Static Transit.). The Feed File should meet following criteria:

1. All of the files in a GTFS feed must be saved as comma-delimited text.
2. The first line of each file must contain field names. Each subsection of the field definitions section corresponds to one of the files in a transit feed and lists the field names that may be used in that file.
3. All field names are case-sensitive.
4. The field values may not contain tabs, carriage returns or new lines.
5. Field values that contain quotation marks or commas must be placed within quotation marks. In addition, each quotation mark in the field value must be preceded with a quotation mark.
6. Remove any extra spaces between fields or field names. Many parsers consider the spaces to be part of the value, which may cause errors.
7. Zip the files in the feed.

Process and Publish the Existing Feeds by Transit Agencies

From Google: Download the existing feeds as a zip file from the Google site, as follows: <https://code.google.com/p/googletransitdatafeed/wiki/PublicFeeds>. After downloading the zip file, extract the .txt files for analysis.

Figure 7 is a list of transit schedule data published by transit agencies and operators in GTFS format for developers to use. They contain scheduled times, stop locations, route and optional fare information, and detailed route shapes.

Google Code Archive

Search this site

Projects Search About

Project **googletransitdatafeed - PublicFeeds.wiki** [Export to GitHub](#)

Source

Issues

Wikis

Downloads

This is a list of transit schedule data published by transit agencies and operators in GTFS format for developers to use. They contain scheduled times, stop locations, route information and optionally fare information and detailed route shapes.

Another [list of official GTFS data](#) is maintained by the [GTFS Data Exchange](#) site.

For details on the feed format, see the [General Transit Feed Specification](#) document.

| Agency | Area | Feed Location | Notes | |-----|:-----|:-----|:-----| | TriMet | Portland, OR | <http://developer.trimet.org/schedule/gtfs.zip> | Use is subject to the terms of their [developer license](#) | | BART | San Francisco Bay Area, CA | <http://www.bart.gov/schedules/developers/open.aspx> | Use is subject to the terms of their [developer license](#). Additional information at <http://www.bart.gov/schedules/developers/open.aspx> | | Caltrain | San Francisco Bay Area, CA | <http://www.caltrain.com/developer.html> | Use is subject to the terms of their [Developer License Agreement](#) | | DART | Dallas, TX | <http://www.dart.org/transitdata/> | Please read [legal notice](#) | | OCTA | Orange County, CA | http://www.octa.net/current/google_transit.zip | Use subject to [disclaimer](#) | | Sacramento RT | Sacramento, CA | <http://portal.sacrt.com/GTFS/SRTD> | License Agreement | | Capital Metro | Austin, TX | http://www.capmetro.org/gisdata/google_transit.zip | Disclaimer and additional files/formats available on their [GIS Data page](#) | | Humboldt Transit Authority | Humboldt County, CA | <http://data.trilliumtransit.com/gtfs/humboldtcounty-ca-us/> | | Rio Vista Delta Breeze | Rio Vista, CA | <http://data.trilliumtransit.com/gtfs/riovista-ca-us/> | | Unitrans | Davis, CA | <http://portal.sacrt.com/GTFS/Unitrans> | License Agreement | | Regional Transportation District | Denver, CO | <http://www.rtd-denver.com/GoogleFeeder/> | License Agreement | | Transperth | Perth, WA (Australia) | <http://www.transperth.wa.gov.au/About/Spatial-Data-Access> | License Agreement | | York Region Transit | York County, ON | http://www.yrt.ca/google/google_transit.zip | | Milwaukee County Transit System | Milwaukee, WI | http://kamino.mcts.org/gtfs/google_transit.zip | Terms of Use | | Oahu Transit Services, Inc | Honolulu, HI | http://webapps.thebus.org/transitdata/Production/google_transit.zip | Terms of Use | | Hampton Roads Transit | Hampton Roads, VA | <http://googlef.gohrt.com/> | | Metrolink | Southern California | http://www.metrolinktrains.com/help/page/title/developer_resources | | Monroe County Transit Authority | Monroe County, PA | <http://www.gtfs-data-exchange.com/agency/monroe-county-transit-authority/> | | Mountain Line | Flagstaff, AZ | <http://data.trilliumtransit.com/gtfs/naipta-az-us/> | | Jefferson Transit Authority | Port Townsend, WA | <http://www.gtfs-data-exchange.com/agency/jefferson-transit-authority/> | License Agreement | | Island Transit | Island County, WA | <http://www.gtfs-data-exchange.com/agency/island-transit/> | License Agreement | | METRO | Houston, TX | <http://www.ridemetro.org/Pages/NewsDownloads.aspx> | Legal Notice | | SFMTA (Muni) | San Francisco, CA | <http://www.sfmta.com/transitdata> | License Agreement | | The Irvine Shuttle | Irvine, CA | http://irvineshuttle.net/files/managed/Document/70/google_transit.zip | Website Policy | | Broward County Transit | Broward County, FL | www.broward.org/bct/google/latest/google_transit.zip | | Blacksburg Transit | Blacksburg, VA | <http://www.gtfs-data-exchange.com/agency/blacksburg-transit/> | | Washington Metropolitan Area Transit Authority | Washington, DC | http://www.wmata.com/rider_tools/developer_resources.cfm | License Agreement |

Figure 7: Google Transit Data Feed

From the GTFS Data Exchange Site: Locate the recent data feeds updates from the Data Exchange at the following link: <http://www.gtfs-data-exchange.com/agencies> (Figure 8). This is another list of official GTFS data maintained by the GTFS Data Exchange site.

Figure 8: GTFS Data Exchange

Understanding GTFS-Realtime

GTFS-realtime is a feed specification that allows public transportation agencies to provide real-time updates about their fleet to application developers. It is an extension of GTFS (General Transit Feed Specification), an open data format for public transportation schedules and associated geographic information. GTFS-realtime was designed around ease of implementation, good GTFS interoperability, and a focus on passenger information. GTFS-realtime provides the status of the fleet, and the feed must be updated regularly, preferably whenever new data comes in from the Automatic Vehicle Location system (AVL).

“The specification was designed through a partnership of the initial Live Transit Updates partner agencies, a number of transit developers and Google. The specification is published under the Apache 2.0 License” (Google Developers. Realtime Transit: <https://developers.google.com/transit/gtfs-realtime/>).

Overview of GTFS-Realtime Feed Types

The feed specification currently supports the following types of information or feed. Updates for each feed type are provided in a separate feed. Feeds are served via HTTP and are updated frequently. The file itself is a regular binary file, so any type of web server can host and serve the file (other transfer protocols can be used as well). Web application servers are also used as an alternative, which if used as a response to a valid HTTP GET request, will return the feed. There are no constraints on the frequency or on the exact method used to update or retrieve the feed.

- **Trip Updates** – Updates on delays, cancellations, and changed routes.
- **Service Alerts** – Alerts that prevent unforeseen events that could affect a station, route or the entire network.
- **Vehicle Positions** – Information about the vehicles, including location and congestion level.

Data Format

The GTFS-realtime data exchange format is based on protocol buffers. Protocol buffers are a language- and platform-neutral mechanism for serializing structured data (like XML, but smaller, faster, and simpler). The data structure is defined in a gtfs-realtime.proto file, which is used to generate source codes to easily read and write the structured data from and to a variety of data streams, using a variety of languages, such as Java, C++ or Python. More information on GTFS-realtime.proto can be found at the following link: <https://developers.google.com/transit/gtfs-realtime/gtfs-realtime-proto>. Additional information on the source code of protocol buffers can be found at <https://developers.google.com/protocol-buffers/docs/downloads>. The overall GTFS-realtime Reference can be found at <https://developers.google.com/transit/gtfs-realtime/reference>

Data Structure

The hierarchy of elements and their type definitions are specified in the *gtfs-realtime.proto* (<https://developers.google.com/transit/gtfs-realtime/gtfs-realtime-proto>) file.

This text file is used to generate the necessary libraries in any programming language. These libraries provide the classes and functions needed for generating valid GTFS-realtime feeds. The libraries not only make feed creation easier, but also ensure that only valid feeds are produced.

GTFS-realtime Term Definition

- **required:** Exactly one
- **repeated:** Zero or more
- **message:** Complex type
- **enum:** List of fixed values

Important Feed Elements

1. message FeedMessage: The contents of a feed message. Table 9 shows a description of the feed header and entity. Each message in the stream is received as a response to an appropriate HTTP GET request. A real-time feed is always defined in relation to an existing GTFS feed. All of the entity IDs are resolved with respect to the GTFS feed. A feed depends on several external configuration factors, as follows:
 - The corresponding GTFS feed.
 - Feed application (updates, positions or alerts). A feed should contain only items from the appropriate applications; all other entities are ignored.
 - Polling frequency, controlled by min_update_delay, max_update_delay.
2. message TripUpdate: Real-time update on the progress of a vehicle along a trip. Depending on the value of ScheduleRelationship, a TripUpdate can specify:
 - A trip that proceeds along the schedule.
 - A trip that proceeds along a route, but has no fixed schedule.
 - A trip that has been added or removed, with regard to schedule.

Table 9: Feed Header and Entity

Field Name	Type	Cardinality	Description
header	FeedHeader	required	Metadata about this feed and feed message
entity	FeedEntity	repeated	Contents of the feed

The updates can be for future, predicted arrival/departure events, or for past events. In most cases, information about past events is a measured value; thus, its uncertainty value is recommended to be zero (0). There are cases, however, when this does not apply because it is allowed to have an uncertainty value different from zero (0) for past events. If an update's uncertainty is not zero (0), either the update is an approximate prediction for a trip that has not ended, the measurement is not precise, or the update was a prediction for the past that has not been verified after the event occurred.

Note that the update can describe a trip that has already ended. Therefore, an update for the last stop of the trip is considered adequate. If the time of arrival at the last stop is in the past, the client will conclude that the entire trip is in the past (it is possible, although inconsequential, to provide updates for preceding stops). This option is most relevant for a trip that has ended ahead of schedule, but according to the schedule, the trip is still proceeding at the current time. Removing the updates for this trip could cause the client to assume that the trip is still occurring. Note that the feed provider to purge past updates is allowed, but not required; this is one case where this would be practical and useful. Table 10 presents a description of some GTFS fields.

Table 10: Example of GTFS Field Information

Field Name	Type	Cardinality	Description
trip	TripDescriptor	required	The trip related to this message.
vehicle	VehicleDescriptor	optional	Additional information on the vehicle that is serving this trip.
stop_time_update	StopTimeUpdate	repeated	Updates to StopTimes for the trip (both future, i.e., predictions, and in some cases, past ones, i.e., those that already happened). The updates must be sorted by stop_sequence and applied for all of the following stops of the trip up to the next specified one.

timestamp	uint64	optional	Moment at which the vehicle's real-time progress was measured in POSIX time (i.e., the number of seconds since January 1, 1970, 00:00:00 UTC).
------------------	--------	----------	--

Non-GTFS Feeds Published by Transit Agencies

A good resource for obtaining non-GTFS feeds published by transit agencies is presented in Figure 9. It also includes known links to GTFS-realtime feeds. The feeds can be viewed in this link: <https://code.google.com/archive/p/googletransitdatafeed/wikis/PublicFeedsNonGTFS.wiki>.

The screenshot shows the Google Code Archive page for the project 'googletransitdatafeed - PublicFeedsNonGTFS.wiki'. The page includes a search bar and a list of project details. The main content is a list of non-GTFS feeds, organized by agency, area, feed location, type, format, and notes. The list includes links to various transit agencies' real-time feeds, such as TriMet, BART, King County METRO, L.A. Metro, Transport for NSW, CTA Bus, SFMTA, ytv, Traveline NextBuses, MBTA/EOT, WMATA, AC Transit, London Datastore, Toronto Transit Commission, NextBus Webservices, DC Circulator, and Dutch Railways.

Figure 9: Agencies with Real-Time Feeds

Combining GTFS Real-Time and Static Data

A diagram was developed by the research team to prepare the GTFS data for the data analysis for this project, as shown in Figure 10. This figure depicts the GTFS static and real-time data elements, its relationships, and the expected outputs. This diagram was used to create the SQL Server database that was used as a base for the data analysis and the HART case study.

Figure 10: Static and Real-Time GTFS Data Diagram

Tri-Met Developer Resources

Another source of information for using GTFS data, which was investigated in this project, was the work and software applications Tri-Met developed and implemented. Tri-Met have made these resources available to software developers in order to promote the use of transit and information related to transit. Tri-Met produces a GTFS feed to share their public transit information with developers, who write tools that consume GTFS feeds to incorporate public transit information into their applications. GTFS can be used to power trip planners, timetable publishers, and a variety of applications.

Static data for Tri-Met can be downloaded from the following links:

<http://www.gtfs-data-exchange.com/agency/Tri-Met/>

<http://developer.Tri-Met.org/schedule/gtfs.zip>.

The Tri-Met Developer Resources, including the GTFS-realtime, can be found at <https://developer.trimet.org/GTFS.shtml>

Tri-Met's GTFS includes a few unofficial GTFS data elements. These elements are documented in Table 11, and Table 12 shows some of the GTFS data elements in Tri-Met's data.

Table 11: Tri-Met's Unofficial GTFS Data Elements

agency.txt		
Field Name	Required	Details
bike_policy_url	Optional	The bike_policy_url specifies the URL of a web page that offers details about the agency's bike policy. The value must be a fully qualified URL that includes http:// or https:// , and any special characters in the URL must be correctly encoded. See https://www.w3.org/Addressing/URL/4_URI_Recommentations.html for a description of how to create fully qualified URL values.
feed_info.txt		
Field Name	Required	Details
feed_id	Optional	The feed_id field contains a universally unique ID for the feed, as found in this link: http://gtfs.org/feeds.txt .
realtime_feeds.txt		
Field Name	Required	Details

url	Required	The url field contains the URL of a GTFS-realtime service.
trip_updates	Required	The trip_updates field contains a binary value that indicates that the URL contains GTFS-realtime trip updates.
alerts	Required	The alerts field contains a binary value that indicates that the URL contains GTFS-realtime service alerts.
vehicle_positions	Required	The vehicle_positions field contains a binary value that indicates that the URL contains GTFS-realtime vehicle positions.
routes.txt		
Field Name	Required	Details
route_sort_order	Optional	The route_sort_order field contains the agency's preferred sort order when displaying route lists (e.g., a route dropdown/combo box).
stop_times.txt		
Field Name	Required	Details
continuous_stops	Optional	<p>The continuous_stops field is used to indicate a section of a trip where it is possible to board or alight from the transit vehicle at any point along the vehicle's path of travel.</p> <p>The field can have the following non-negative integer values:</p> <ul style="list-style-type: none"> • 0 or blank - Normal stop behavior along route (default). • 1 - Continuous stopping behavior from this stop-time to the next stop-time in the trip's sequence.

Table 12: Commonly Used Transit Terms Used in Tri-Met's Data and their Definitions

Block	A collection of trips assigned to a vehicle for a day. An older term for this is "Train," which is still used internally by Tri-Met.
Direction	One of two possible directions of travel--inbound and outbound--on a route. (Example: Line 17-Holgate is "inbound" to Sauvie Island, and "outbound" to 136th and Powell.)
Location ID	The unique number that identifies each bus and MAX stop. In public viewing, this number is referred to as a StopID.
Pattern	A unique travel path through the street network to visit selected stops for one trip.
Route	A collection of ordered stops presented publicly as a "line" with a name (such as "17-Holgate") and represented internally as an integer.

Timepoint	One of several stops on each route that serves as a benchmark to show whether a trip is running on time.
Train	An older term for “Block,” which is still used internally by Tri-Met. The terms are interchangeable.
Trip	An iteration of a vehicle traveling along a given pattern. (Example: Line 17 makes a trip from downtown to 136th and Powell. The next trip would be 136th and Powell to downtown.)

Tools developed using Tri-Met's developer resources include Acehopper, Arrival, ArrivalTracker, Dadnab, and efoBus, as shown in Figure 11.

[Learn](#)
 Fares, How to Ride
 Bikes, Parking, Rules...

[Buy](#)
 Tickets, Passes
 Gear, Apps...

[Contact](#)
 Customer Service
 Lost & Found...

[Email](#)
[Tweet](#)
[Share](#)

TriMet App Center

Transit tools for the web and mobile devices

Below are some of the free and commercial applications that are available from third-party developers using TriMet's [open data](#).

Show only:
Showing 56 of 56 apps

☐ Android
 ☐ iPhone
 ☐ iPod Touch
 ☐ iPad
 ☐ Blackberry

☐ Windows Phone
 ☐ Mobile Web
 ☐ Web
 ☐ Text Messaging
 ☐ Other

Acehopper
Provides schedule and real-time information.
For Android, iPhone, iPod

Arrival
Provides real-time arrival information.
For iPhone, iPad

ArrivalTracker
Searches for nearest stops, provides real-time arrival information.

Dadnab™
Provides public transit directions (trip planning) via text messaging.

efoBus
Searches for stops, provides arrival information, displays a

Figure 11: Tri-Met Developer Resources

Tri-Met Real-Time Application Programming Interface (API)

In addition to a GTFS-realtime feed, Tri-Met also provides a web service in a proprietary format. This API pre-dates the GTFS-realtime format, and was the primary method of sharing real-time data prior to the existence of GTFS-realtime. All of Tri-Met's web services are read-only in nature, and use HTTP or HTTPS as the transport mechanism with HTTP GET as the method to call the service, resulting in XML formatted data as the response. Each web service begins with a base URL, followed by parameters and arguments. The service arguments are separated by either a forward slash ("/") as part of the URL path, or with GET parameters with the standard "?/&/" separators.

An application ID (AppID) is a required parameter for all Tri-Met web service calls. Therefore, before using the services, a user must register for an AppID. An AppID can be created at the following link: <https://developer.Tri-Met.org/appid/registration/>. Figure 12 depicts a screenshot of the registration page.

The screenshot shows the 'Application registration' page on the Tri-Met Developer Resources website. The page has a blue header with the 'Developer Resources' text and the Tri-Met logo. Below the header, a notice states: 'All information will be treated in accordance with our [privacy policy](#).' The main content area is titled 'Application registration' and contains several form fields: 'Contact name:' with a text input, 'Email address:' with a text input, 'Year and month of birth:' with dropdown menus for '2015' and 'June', 'Application name:' with a text input, and 'Company name (optional):' with a text input. Below these is a large text area for 'Notes or Comments (optional):'. To the right of the form is a sidebar titled 'DEVELOPER RESOURCES' with a list of links: 'Home', 'GTFS & GTFS-realtime', 'GIS Data', 'TimeTable Publisher', 'Privacy Policy', 'Registration', 'Terms of Use', 'Transit Terms', 'Why Register?', and 'Web Services Documentation'. At the bottom of the form is a reCAPTCHA image showing a mailbox with the number '149' and a text input field with the placeholder 'Type the text'. Below the text input is a link for 'Privacy & Terms'.

Figure 12: Tri-Met's Registration Page

Example Service Calls

From the Tri-Met website, the following are examples of service calls:

First, slashes are used to separate the service parameters:

<https://developer.trimet.org/ws/V1/arrivals/locIDs/6849,6850/appID/000000000000000000000000>

Second, the HTTP GET parameters style is used, as follows:

<https://developer.trimet.org/ws/V1/arrivals?locIDs=6849,6850&appID=000000000000000000000000>

There are various available web services provided by Tri-Met, as shown in Table 13.

Table 13: Tri-Met Web Services

BETA: Vehicles	Latest vehicle positions. (https://developer.Tri-Met.org/ws_docs/vehicle_locations_ws.shtml)
BETA: Arrivals V2	Reports arrivals at a stop identified by location ID. (https://developer.Tri-Met.org/ws_docs/arrivals2_ws.shtml)
Arrivals	Reports next arrivals at a stop identified by location ID. (https://developer.Tri-Met.org/ws_docs/arrivals_ws.shtml)

As mentioned earlier, all of Tri-Met's web services use HTTP GET as the method to call the service and use XML formatted data as the response. XML data obtained from the response are in the following format (shown in Figure 13).

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0"?>
<resultSet xmlns="urn:trimet:arrivals" queryTime="1433745097469">
  <location desc="SE 17th & Center" dir="Southbound" lat="45.4941036890001" lng="-122.648671236999" locid="6850"/>
  <arrival block="1704" departed="true" dir="0" status="estimated" estimated="1433746283000" fullSign="17 Holgate to SE 134th Ave" piece="1" route="17" scheduled="1433746200000" shortSign="17 To SE 134th" locid="6850" detour="true">
    <blockPosition feet="26450" at="1433745063000" heading="270" lat="45.5350999" lng="-122.6660861">
      <trip desc="Holgate & SE 134th Dr" dir="0" route="17" tripNum="5506954" destDist="49642" pattern="26" progress="23192"/>
    </blockPosition>
  </arrival>
  <arrival block="1703" departed="false" dir="0" status="scheduled" fullSign="17 Holgate to SE 134th Ave" piece="1" route="17" scheduled="1433749320000" shortSign="17 To SE 134th" locid="6850" detour="true"/>
  <arrival block="7003" departed="false" dir="1" status="scheduled" fullSign="70 12th Ave to Milwaukie via 17th & Tacoma" piece="1" route="70" scheduled="1433768400000" shortSign="70 To Milw via 17" locid="6850" detour="true"/>
</resultSet>
```

Figure 13: Screenshot of Tri-Met XML Data Responses

This XML response requires data processing, such as providing styling information and removing the ampersand (&) sign.

Process of Storing XML Responses at the Local Machine

The XML data can be manually stored on a local machine; however, this method is a time-consuming process. In addition, the XML response is dynamic in nature. This problem is solved with the use of C#.net, which automates the process, (Figure 14).

Figure 14: Process of Storing Tri-Met XML Data Responses

Methods Used in Storing Data

In the abovementioned C#.net code, there is a page load event. When the page loads, it will run an infinite loop. In that loop, the webClient “method of” is used. The desired URL is opened and saved with an .XML format and a timestamp. Since the data is dynamic, a delay of 30 seconds is applied. After 30 seconds, the data is refreshed and is stored as described earlier.

Timestamps in XML Response

The XML response contains time in epoch time. The UNIX epoch (or UNIX time or POSIX time or UNIX timestamp) is the number of seconds that have elapsed since January 1, 1970 (midnight UTC/GMT), not counting leap seconds (in ISO 8601: 1970-01-01T00:00:00Z). The epoch is UNIX time 0 (midnight 1/1/1970), but “epoch” is often used as a synonym for “UNIX

time.” Many UNIX systems store epoch dates as a signed 32-bit integer, which might cause problems on January 19, 2038 (known as the Year 2038 problem or Y2038).

A simple calculation for epoch to human readable format is:

$$\text{Actual time} = (A/86400) + 25569 + (-5/24), \text{ where } A = \text{time in epoch.}$$

Saving XML Response Data to the SQL Server

After saving the data as an XML response, as shown in Figure 15, save the data into the SQL server to perform several actions.

Following are steps using the SQL server 2008.

STEP 1 - In SQL server 2008, create tables (with desired columns as in XML) analogous to the structure of XML.

STEP 2 - In Visual Studio, create a project in C#.net.

STEP 3 - Add an ASP.net web page. In the Page Load event, write code.

STEP 4 - Add XML to the Project.

STEP 5 - Specify a connection string in the web config file for connectivity with the SQL server.

STEP 6 - Using the SQLBULKCOPY class, insert the data into the SQL server.

STEP 7 - Save the project in Visual Studio and build it.

STEP 8 - After running the project, data from XML are inserted into the SQL server.

STEP 9 - Using the SELECT command, the changes can be seen in the SQL server.

Figure 15: Screenshot of Saving Tri-Met XML Data Responses to SQL Server

Other Considerations

This section presents some basic information on the calculation of geographical distances and software development matters that can help with the use of GTFS data and with the potential development of a software application.

GPS Coordinates: Latitude and Longitude

(MapTools. Tools for Navigation: Latitude and Longitude definitions. https://www.maptools.com/tutorials/lat_lon/definitions)

Latitude

Lines of latitude measure the north and south positions between the poles. The equator is defined as 0 degrees, the North Pole is 90 degrees north, and the South Pole is 90 degrees south. Lines of latitude are all parallel to each other, thus they are often referred to as parallels (Figure 16).

Figure 16: Latitude

One degree of latitude is 60 nautical miles, 69 statute miles or 111 km.

One minute of latitude is 1 nautical mile, 1.15 statute miles, or 1.85 km.

Longitude

Lines of longitude, or meridians, run between the North and South Poles and measure east-west positions (Figure 17). The prime meridian is assigned the value of 0 degrees, and runs through Greenwich, England. Meridians to the west of the prime meridian are measured in a number of degrees west, and likewise, those to the east of the prime meridian are measured by their number of degrees east.

Figure 17: Longitude

Latitude and Longitude Formats

There are a variety of formats for latitude and longitude, for example:

- deg-min-sec suffixed with N/S/E/W (e.g. 40°44'55"N, 73 59 11W), or
- signed decimal degrees without compass direction, where negative indicates west/south (e.g. 40.7486, -73.9864).

Distance Calculation

Distance is calculated using a set of latitude and longitude pairs. In this project, a variation of the “haversine” formula is used to calculate the great-circle distance between two points; that is, the shortest distance over the earth’s surface. The haversine formula is an equation, which provides the distances between two points on a sphere from their longitudes and latitudes.

Let us consider:

I2= latitude 1 in decimal

J2 = longitude 1 in decimal

I3 = latitude 2 in decimal

J3 = longitude 2 in decimal

D = distance between the points

Then, the formula will be:

$$D = 6371 * \text{ACOS}(\text{COS}(\text{RADIANS}(90-I2)) * \text{COS}(\text{RADIANS}(90-I3)) + \text{SIN}(\text{RADIANS}(90-I2)) * \text{SIN}(\text{RADIANS}(90-I3)) * \text{COS}(\text{RADIANS}(J2-J3))) / 1.609$$

The haversine formula remains particularly well-conditioned for numerical computation, even at large distances, as well as for small distances, unlike calculations based on the spherical law of cosines (Movable Type Scripts Haversine’s formula: <http://www.movable-type.co.uk/scripts/latlong.html>)

Advantages of using the Haversine Formula

The technology for a navigator’s calculations used to be log tables. As there is no real log of a negative number, the “versine” enabled them to keep trigonometric functions in positive numbers. Also, the $\sin^2(\theta/2)$ form of the haversine avoided addition, which entailed an anti-log lookup, the addition, and a log lookup. Printed tables for the haversine/inverse-haversine and its logarithm that was used to aid multiplications saved navigators from squaring sines, computing square roots, etc., which was an arduous and error-prone activity. This is the biggest advantage of using the haversine formula.

Limitation of using Haversine Formula

Since this formula involves calculating the shortest distance between two points, this formula is useful if a route pattern of a bus is a straight line or an approximately straight line. For instance, case A in Figure 18 shows straight or approximately straight lines. Therefore, this case is useful for calculating distances using the haversine formula. On the other hand, if case B is considered,

the route pattern of the bus is not straight. If the distance is calculated using the haversine formula, the result is incorrect because it does not follow the path of the route.

A

B

Figure 18: Route Patterns

Software Development

Due to the complexity of using GTFS and AVL data, the ideal solution is to develop a computerized tool to assist transit agencies in improving transit service reliability. With a computerized system, management and staff can easily monitor, manage, and improve the efficiency of the transit system. This section presents some software development considerations in the development of a Transit Reliability System.

Software Design Description (SDD)

Software Design Description (SDD) defines and describes the use of each view, the functional requirements with a significant impact on the architecture, use-case realization and the layers and subsystems of the application.

SDD within the Life Cycle

In the design element of the Software Development Life Cycle (SDLC), as shown in Figure 19, the software design description is defined. SDD is a written description of a software and provides overall guidance to the architecture of the software project. As the cycle advances through the design cycle, and after the requirements and specifications are gathered, a prototype of the software application is designed. Before implementation, it is crucial to design the software as more controlled in the coding part. As seen in Figure 19, if there is a problem in the testing or maintenance process, the design needs to be reviewed and the code fixed.

Figure 19: Software Development Life Cycle (SDLC)

(Source: <http://2010.igem.org/Team:Newcastle/E-Science>)

Purpose of an SDD

The main purpose of this SDD is to define and describe the use of view, the functional requirements with a significant impact on the architecture, use-case realization and the layers and subsystems of the application.

Architecture

This section is based on the Software Engineering Standards Committee of the IEEE Computer Society, IEEE Recommended Practice for Software Design Descriptions, and IEEE Standard 1016-1998. The Software Design Descriptions (SDD) provides an architectural overview of the Transit Reliability System (Figure 31).

Scope: The scope of this SDD is to define a high-level design of the Transit Reliability System.

Audiences: Management and transit agency staff.

Technical Platform: Software is deployed onto any application server.

Security: The system must be secured.

Basic security behaviors:

- **Authentication:** Login using a user name and a password.
- **Authorization:** According to their profile, an online user must be granted or not allowed to receive some specific services.

For internet access, the following requirements are mandatory:

- **Confidentiality:** Sensitive data must be encrypted, if any.
- **Data integrity:** Data sent across the network cannot be modified by a tier.
- **Auditing:** Every sensitive action can be logged.

Persistence: Data persistence will be addressed using a relational database.

Reliability/Availability: High availability is required since there are monitory issues related to the system's availability. The system's high availability will also ensure system efficiency.

Performance: Search queries should return 90% of the time below 5 seconds. Since this is a monitory system, the response of the system should be fast.

Figure 20 displays a view of the software architecture of a potential Reliability System that can be used to improve on-time performance and monitor bus bunching. In this diagram, on-time performance and bus bunching are presented as a means to improve transit reliability. For both cases, a system of alarms could be developed to assist transit staff, based on different conditions.

Figure 20: Architectural Overview of a Reliability System

In this architectural view, there are two components of the system: On-Time Performance (OTP) and Bus Bunching. In the first component, there are three sub-components: early, on time, and late. Feedback is sent to the system with any of the responses (early, on time, and late).

The alarms could be set up on stages based on the severity and the level of detail (e.g., trip, route, or system level). For instance, for on-time performance, an alarm for Stage 3 could be set up to be lower than 50% for a trip. This can alert an operation supervisor to take immediate actions before the service deteriorates any further. Ideally, the issue should be addressed at Stage 1, when OTP is set up to be at 70%. Similarly, bus bunching is addressed based on the headway between vehicles. That is, if the actual headway is half of the scheduled headway, the Stage 2 alarm is triggered. In any case, the settings should be based on agency policy and availability of resources.

In the Use Case depicted in Figure 21, the system will be used by transit agency staff that can perform the following actions:

1. Report Generating
2. Measure OTP and Bus Bunching
3. Monitoring
4. Intelligent Decision
5. Planning & Scheduling

Figure 21: Use Case

The reliability system can be divided into layers based on the N-tier architecture (Figure 22).

Figure 22: N-Tier Architecture

The layering approach is the most accepted solution for enterprise applications, which require scalability, modularity and easy maintenance.

- The web tier deals with the presentation logic and the page rendering.
- The business tier deals with the core functionalities of the system. (Search, post, match rides, manage profiles)
- The EIS tier is responsible for storing user profiles and critical data.

CASE STUDIES

In order to better present the ideas explored in this research project, two transit agencies were used as case studies: Hillsborough Area Regional Transit Authority (HART) and Miami-Dade Transit (MDT). Furthermore, Route 5 from HART, and Routes 288 and 11 from MDT were selected for the data analysis task. Figures 23, 24, and 25 show the bus routes from HART and MDT. The HART Route 5 extends from the University of South Florida (USF) to Downtown Tampa.

Figure 23: HART Transit Route 5

MDT Route 288 goes from the Dadeland North Metrorail Station to the West Kendall Transit Terminal, and Route 11 goes from downtown Miami to Florida International University (FIU).

Figure 24: Miami-Dade Transit Route 288

Figure 25: Miami-Dade Transit Route 11

The main objective for analyzing the data from these case studies is to demonstrate how transit reliability is improved by modifying timetables and monitoring schedule adherence and bus bunching.

On-Time Performance Analysis

Data preparation

For this section, AVL data from MDT Route 11 is used. From the three months (August-October, 2015) of archived data that covered the entire MDT system, the records for route 11 were queried and exported. This query returned a total of 61,629 records. Unnecessary columns for this research were deleted to decrease the volume of data. The following columns were removed from the dataset:

- ID
- Distance Traveled
- Total Stops
- Previous Time Point Sequence
- Previous Time Point Stop ID
- Previous Time Point Stop Name
- Previous Time Point Distance Traveled
- Previous Time Point Departure Time
- Running Time
- Running Time Distance

The following columns are the remaining fields used in the data analysis:

1. Route ID
2. Route Alias
3. Direction Name
4. Service Name, Pattern
5. Run Number, Trip ID
6. Destination
7. Bus Name
8. Stop ID
9. Stop Name
10. Sequence
11. Scheduled Time
12. Arrival Time
13. Departure Time
14. Dwell Time
15. Adherence

The most important column in this study is Adherence (schedule adherence measured in seconds). This variable shows how much earlier or later from the scheduled time a bus departed from the stop location.

Different on-time performance criteria are introduced, but the most commonly used parameters are presented below (Cevallos, 2011). Buses are considered on time when they are within:

- a. 5 minutes late and 2 minutes early [-5, 2].
- b. 5 minutes late and 1 minutes early [-5, 1].
- c. 5 minutes late and 0 minutes early [-5, 0].

Note that depending on the AVL system, the signs for early and late parameters can be the opposite. For example, instead of being [-5, 2], it could be [-2, 5]. That is, 2 minutes early will be noted as -2 and 5, minutes late will be 5. The results are the same, and only the sign changes. This is due to the calculation of real-time departure time – scheduled time against scheduled time - real-time departure time.

Another issue to consider is the use of arrival time versus departure time; agencies may decide to use either one for different reasons. However, for measuring on-time performance, the use of departure time is preferred. For instance, a bus could arrive early, but depart on time. In any case, buses should not depart early.

Furthermore, to ensure that the data did not include extreme values commonly associated with anomalies, the outliers were removed from the dataset using the boxplot method. Table 14 shows the values used in the boxplot method to remove the outliers from the Route 11 data.

Table 14: Box Plot Method Applied to Remove Outliers (Adherence Variable)

Min.	1st quartile	Median	Mean	3rd quartile	Maximum
-4482	-537	-242	-375	-76	1048

Based on the data in Table 14, the average number of buses are arriving 375 seconds (6.25 minutes) later to the stop location. The following methodology is used to remove the extreme outliers:

$$IQR = -76 - (-537) = 461$$

$$Higher\ boundary = 3 * 461 - 76 = 1307$$

$$Lower\ boundary = -537 - 3 * 461 = -1920$$

The formula above shows the data in which their adherence variable is either less than 1920 or greater than 1307, which should be removed from dataset. After removing the outliers, 60,808 records are available.

Fitting Distributions

To analyze the on-time, early, or late arrival of buses, a histogram of the adherence variable was used. The histogram is the graphical representation of a numerical value; however, to study the variable continuously, a probability distribution curve can be fitted to the existing histogram. The next section shows the different distributions, which were assigned to four levels of data, including route level, direction level (east and west), and a specific trip level. Minitab© software is used in this study to find the best available distribution curve for each histogram (Arend, 1993).

For this analysis, 60,808 data records from Route 11 were used. Figure 26 shows the histogram of this data.

Figure 26: Scheduled Adherence Histogram for Route 11

Table 15 displays the data derived from the use of the distribution identification tool in the Minitab© software.

Table 15: Distributions Analysis for Route 11

Distribution	AD	P-Value	Location	Scale	Threshold
Normal	2033.594	<0.005	-5.78498	6.43278	
Three-Parameter Lognormal	2047.028	<0.010	7.10118	0.00532	-1219.18072

Smallest Extreme Value	435.335	<0.010	-3.01643	4.54829	
Largest Extreme Value	4456.845	<0.010	-9.35726	8.21975	

According to Table 15, two distributions of Smallest Extreme Value and Largest Extreme Value are selected and fitted into the data. In addition, the Three-Parameter Lognormal and Normal distributions were included, as they are commonly used in this type of analysis. From these distributions, the Smallest Extreme Value is the best fit, based on the p-value and the Anderson-Darling statistic (AD). The fitted distributions are shown in Figures 27 to 30.

Figure 27: Normal Distribution Fitted on Route 11 Schedule Adherence Histogram

Figure 28: Smallest Extreme Value Distribution Fitted on Route 11 Data

Figure 29: Largest Extreme Value Distribution Fitted on Route 11 Data

Figure 30: 3-Parameter Lognormal Distribution Fitted on Route 11 Data

As shown in Figures 27 to 30, the smallest extreme value distribution with a location parameter of -3.016 and a scale parameter of 4.548 is the best fit for Route 11 data. The location and scale parameters in the smallest extreme value distribution are considered representative of the mean and standard deviation in the normal distribution. The difference is that the mean and standard deviation for normal distributions are based on the arithmetic mean and standard deviation. However, the location and scale parameters, which are defined for the Lognormal, Weibull and Extreme Value Distributions, use the geometric mean and standard deviation.

Smallest Extreme Value Distribution

Three types of asymptotic distributions were developed for maximum and minimum values based on different initial distributions. These distributions are based on the extreme types theorem, and are widely used in risk management, finance, economics, material science and other industries. Two of these distributions are of particular interest in reliability engineering: the type I asymptotic distribution for both maximum and minimum values (for minimum values, referred to as the Gumbel/Smallest Extreme Value (SEV) distribution in Weibull++), and the type III asymptotic distribution for minimum values (i.e., the well-known Weibull distribution).

The extreme value type I distribution has two forms. One is based on the largest extreme, and the other is based on the smallest extreme. These two forms can be used to model the distribution of

the maximum or minimum number of the samples of various distributions. The general formula for the *pdf* of the type I (minimum) distribution is:

$$f(x) = \frac{1}{\sigma} e^{\frac{x-\mu}{\sigma}} e^{-e^{\frac{x-\mu}{\sigma}}}$$

where:

- ✓ μ is the location parameter.
- ✓ σ is the scale parameter.

When $\mu=0$, $\sigma=1$, the above equation reduces to the standard Gumbel (minimum) distribution:

$$f(x) = e^x e^{-e^x}$$

The reliability function of the Gumbel (minimum) distribution is given by:

$$R(x) = e^{-e^x}$$

Direction Level (Eastbound of Route 11)

Using the same procedure as the previous section, the data for the eastbound direction of Route 11 is fitted. The distribution was based on 32,111 records, and similar to the entire Route 11, the smallest extreme value distribution was the best fit with a locations parameter of -2.689 and scale parameter of 4.115.

Figure 31: Smallest Extreme Value Distribution Fitted on Route 11 (EB) Data

Direction Level (Westbound of Route 11)

The Westbound direction of Route 11 was composed of 28,697 records. The smallest extreme value distribution was the best fit with a locations parameter of -3.416 and scale parameter of 5.052.

Figure 32: Smallest Extreme Value Distribution Fitted on Route 11 (WB) Data

Trip Level

To show the analysis at the trip level, Trip 3491000, which runs in the afternoon peak hour, was selected. A total of 259 data records were obtained for this specific trip. Analysis showed a smallest extreme value distribution, with -1.238 as the location parameter and 2.17 as the scale parameter.

Figure 33: Smallest Extreme Value Distribution Fitted on Trip 3491000

On-time Performance at the Timepoint Level

In this section, the on-time performance of two trips (Trip 3491000 and Trip 3491129) in the eastbound and outbound direction are analyzed (Table 16). As shown in Table 16, the on-time performance parameters were considered and labeled as Cases 1, 2 and 3. The parameters are the equivalent to the on-time performance measure of 5 minutes late and 2 minutes early (Case 1), 5 minutes late and 1 minute early (Case 2), and 5 minutes late to 0 minutes early (Case 3). In addition, the on-time percentages vary among timepoints along the trips, as well as per direction. This variability can be reduced by proper scheduling and real-time operation control. Trip 3491129 is a case of a low OTP, which cannot be improved only by scheduling techniques. Real-time monitoring can be used to keep buses under the agency's on-time parameters, which can improve OTP and reduce the number of late and early events.

According to these assessments, using different parameters can have a significant impact on the on-time performance outcome. Since transit agencies use different parameters, the OTP comparison among agencies is difficult. Therefore, the need for standardized OTP parameters is crucial for the comparison of service reliability across different agencies.

Table 16: On-Time Performance Measure Percentile for Two Sample Trips**Route 11**

Eastbound				Westbound			
Trip ID : 3491000				Trip ID: 3491129			
Stop	Case 1	Case 2	Case 3	Stop	Case 1	Case 2	Case 3
1 (Stop ID 100)	92	75	50	1 (Stop ID 104)	44	44	42
2 (Stop ID 101)	78	75	65	2 (Stop ID 106)	40	40	37
3 (Stop ID 102)	80	78	59	3 (Stop ID 107)	44	44	40
4 (Stop ID 103)	67	64	46	4 (Stop ID 108)	45	45	43
5 (Stop ID 104)	92	92	88	5 (Stop ID 952)	56	56	49
6 (Stop ID 961)	88	79	58				
7 (Stop ID 1215)	87	87	83				
8 (Stop ID 1271)	91	91	77				

On-Time Performance Optimization (For Trip 3491000)

Based on Table 16, some stops (timepoints) show acceptable on-time performance, while some stops show very low percentage rates of on-time performance. To find an optimized adherence value, these values are changed gradually (in 1-minute intervals) to maximize the percentage of on-time performance. New on-time performance values for the eastbound direction of Trip 3491000 are shown in Table 17.

Table 17: On-Time Performance Value Adjustment

Timepoints (Stop ID)	100	101	102	103	104	961	1215	1271
Adjustment								
-2 minutes	81	58	66	46	52	82	52	50
-1 minute	83	68	76	56	75	85	83	68
0 (base scenario)	92	78	80	67	92	88	87	91
1 minute	78	75	90	72	96	82	96	95
2 minutes	56	78	71	67	92	61	91	82

For trip ID 3491000, there are 153 records, after removing the end-of-line timepoints (Stops 100 and 1271). Based on Table 17, the schedule adherence was optimized and reconstructed a new curve, as shown in Figure 34. Table 18 shows the adjustment value. Since stop IDs 101 and 961 showed best on-time performance for their existing condition, their adjustment value is 0, and other stops showed +1 as the best adjustment that improved on-time performance.

Table 18: Adjustment Value for Existing Adherence in Trip 3491000

Timepoints (Stop ID)	Adjustment value to existing adherence
101	0
102	+1
103	+1
104	+1
961	0
1215	+1

Figure 34 shows the smallest extreme distribution curve on the adjusted histogram for Trip 3491000. To obtain this histogram, +1 minute is added to the adherence values of all values for Trip 3491000 with stop IDs 102, 103, 104 and 1215. In addition, the original adherence values of all values for Trip 3491000 with stop IDs 101 and 961 are kept, since no improvement occurred when changing the adherence value. Figure 35 shows the before and after adjustment distribution curves.

Figure 34: Smallest Extreme Value Distribution Fitted on Adjusted Trip 3491000

Although Trip 34910000 had reasonably good OTP (80%), it could still improve 1% by using scheduling strategies and adjusting the times at the timepoints, as shown in Figure 35. Further improvements can be achieved by using real-time operational strategies to reduce the late and early arrival/departures.

Figure 35: Before/After Adjustment Distribution Plots for Trip 3491000

Table 19 shows the statistical parameters for the two sets of data.

Table 19: Before/After Adjustment Data Statistical Parameters

Dataset \ Source	Histogram		Smallest Extreme Value Distribution	
	Mean	Std. Dev	Mean	Std. Dev
Before adjustment	-2.404	2.947	-1.275	2.287
After adjustment	-2.137	3.076	-0.7423	2.34

Figure 36 shows a bar chart for adherence values before and after the adjustment.

Figure 36: Before and After Adjustment at the Timepoint Level

Figure 37 shows an example of what would happen if the variability of adherence values were reduced. This figure clearly shows that OTP is optimized, as most of the area under the SEV distribution is within the OTP parameters.

Figure 37: Before/After Distributions with Variability Adjustments for Trip 3491000

On-Time Performance Optimization (For Trip 3491108)

In this section, the trip optimization procedure is used for a trip with a lower on-time performance (Trip 3491108) (Table 20). The average on-time performance for this trip is calculated as 54%. This is a weekday trip, which is normally conducted around 8:00 a.m. Stops are 104, 106, 107, 108, 453, 952, and 1215.

Table 20: On-Time Performance for Trip 3491108

Route 11	
Trip ID : 3491108	
Timepoints	Case 1
1 (Stop ID 104)	66 %
2 (Stop ID 106)	57 %
3 (Stop ID 107)	53 %
4 (Stop ID 108)	48 %
5 (Stop ID 453)	47 %
6 (Stop ID 952)	60 %
7 (Stop ID 1215)	23 %

In order to find an optimized adherence value, the value is changed gradually (in 1-minute intervals) to maximize the percentage of on-time performance for each timepoint on the trip. The adjusted on-time performance values are shown in Table 21, and the number of minutes each timepoint needs adjusting is displayed in Table 22.

Table 21: On-Time Performance Value Adjustment

Timepoints (Stop ID)	104	106	107	108	453	952	1215
Adjustments							
-2 minutes	39%	26%	33%	31%	37%	44%	15%
-1 minute	63%	40%	42%	40%	39%	52%	23%
0 (base scenario)	66%	57%	53%	48%	47%	60%	23%
1 minute	71%	66%	67%	60%	43%	66%	23%
2 minutes	79%	74%	64%	65%	43%	64%	31%
3 minutes	76%	71%	62%	60%	43%	68%	38%
4 minutes	68%	60%	58%	58%	41%	44%	38%
5 minutes	53%	60%	47%	48%	43%	36%	38%
6 minutes	32%	46%	38%	40%	41%	32%	31%

Table 22: Adjustment Value for Existing Adherence in Trip 3491108

Stop ID	Adjustment value to existing adherence
104	+2
106	+2
107	+1
108	+2
453	0
952	+3
1215	+3

Figure 38 shows the smallest extreme distribution curve on the adjusted histogram for Trip 3491108. To obtain this histogram, the existing schedule adherence records were modified using the adjustment values in Table 22 to generate a new dataset. Figure 39 shows the before and after adjustment distribution curves.

Figure 38: Smallest Extreme Value Distribution Fitted on Adjusted Trip 3491108

The use of the area under the Smallest Extreme Value Distributions within the [-5, 2] OTP parameters yielded an increase after the timepoint adjustments. For instance, in Figure 39, the on-time area increased from 46% to 53%, which is an approximate increase of 7% for that particular trip.

Figure 39: Before/After Adjustment Distribution Plots for Trip 3491108

Table 23 shows the statistical parameters for the two sets of data.

Table 23: Before/After Adjustment Data Statistical Parameters

Dataset \ Source	Histogram		Smallest Extreme Value Distribution	
	Mean	Std. Dev	Location	Scale
Before adjustment	-6.32	6.38	-3.64	4.197
After adjustment	-4.61	6.44	-1.904	4.265

Figure 40 shows an OTP bar chart for all of the timepoints along Trip 3491108, both before and after the time adjustments.

Figure 40: Before and After Adjustment at the Timepoint Level

Figure 41 demonstrates the OTP improvements by keeping the location parameter as constant, while changing the scale parameter. As the scale parameter relates to the variation of schedule adherence values, improving OTP can be achieved by the use of good operational strategies, such as monitoring of the transit fleet and taking immediate action to prevent service degradation. For instance, if the scale value changed from 4.265 to 0.865, the area under the curve of the distribution increases from 53% to 97% (44% improvement), which is a considerable increase. This is only obtained when strict operational procedures are in place.

Figure 41: Before/After Adjustment of Scale Parameter for Trip 3491108

Finally, Figure 42 shows an example of further OTP improvements. In this case, the scale parameter is kept constant, and the location parameter is shifted to maximize the area within the on-time performance parameters. As a result, a 99.71% OTP is achieved (2.71% improvement). This fictitious exercise is for illustration purposes and can only be achieved by an exceptional combination of operations strategies (control of variability) and scheduling techniques (modification of the timetables).

Figure 42: Before/After Adjustment of Location Parameter for Trip 3491108

Bus Bunching Analysis

In public transportation, bus bunching is a condition that occurs when buses traveling in the same direction draw closer to one another in terms of distance, so that eventually they travel in a bunch or group along the route. This occurs when at least one of the vehicles is unable to maintain its schedule and therefore ends up closer to the other vehicles. Bus bunching results in unreliable service, long wait times for passengers, overcrowded vehicles, and near empty buses.

Prevention of Bus Bunching

Bus bunching can be prevented in several ways. Following are some commonly used strategies to prevent bunching.

1. Hold buses at certain timepoints along the route.
2. Allow crowded buses to skip certain stops.
3. Maintain a maximum number of passengers in the bus.

Calculation of Bus Bunching

Our approach for calculating bus bunching is based on time-space diagrams of the trips along a route. For the data analysis, the HART General Transit Feed Specification (GTFS) data was stored in an SQL Server database, and Route 5 was used for the analysis and the creation of the figures.

Following are the basic steps for calculating bus bunching:

1. Collecting data for a route.

The data is queried for an entire route for a specific time period using an SQL statement for the combined GTFS static and GTFS real-time tables.

```
select  T.trip_id,T.direction_id,  STU.stop_id,STU.stop_sequence,S.stop_lat,S.stop_lon
,STU.arrival_delay,TU.timestamp,  VP.position_latitude,VP.position_longitude  from
Trips T,trip_updates TU, stop_time_updates STU, vehicle_positions VP, stops S where
T.route_id=5  and  TU.trip_id=T.trip_id  and  S.stop_id=STU.stop_id  and
STU.trip_update_id=TU.oid  and  (VP.vehicle_id=TU.vehicle_id  and
VP.timestamp=TU.timestamp)
```

Figure 43 shows the tables used in the SQL query.

Figure 43: Static and Real-Time GTFS Tables

The dataset results in the Excel spreadsheet is the collection of all trips, which belongs to Route 5, as shown in Figure 44.

Route 5.xlsx - Excel

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW DEVELOPER PDF

Clipboard Font Alignment Number Styles Cells Editing

M16

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	trip_id	direction	stop_id	stop_sequence	stop_lat	stop_lon	arrival	timestamp	position	position_longitude					
2	42275	0	6968		1	27.95476	-82.4583	240	12/7/2014 12:43	27.96228	-82.45652771				
3	42275	0	6968		1	27.95476	-82.4583	180	12/7/2014 12:46	27.96231	-82.44089508				
4	42275	0	6968		1	27.95476	-82.4583	180	12/7/2014 12:47	27.96241	-82.43519592				
5	42275	0	6968		1	27.95476	-82.4583	180	12/7/2014 12:48	27.96412	-82.43473053				
6	42275	0	6968		1	27.95476	-82.4583	180	12/7/2014 12:49	27.96657	-82.43475342				
7	42275	0	6968		1	27.95476	-82.4583	180	12/7/2014 12:50	27.96677	-82.43087769				
8	42291	1	1616		16	28.03471	-82.4164	120	12/7/2014 12:50	28.03264	-82.41641998				
9	42291	1	1616		16	28.03471	-82.4164	120	12/7/2014 12:51	28.03032	-82.41644813				
10	42291	1	1616		16	28.03471	-82.4164	180	12/7/2014 12:51	28.03032	-82.41644287				
11	42291	1	1616		16	28.03471	-82.4164	180	12/7/2014 12:52	28.02459	-82.41634369				
12	42275	0	6968		1	27.95476	-82.4583	180	12/7/2014 12:53	27.9688	-82.4223175				
13	42291	1	1616		16	28.03471	-82.4164	120	12/7/2014 12:53	28.0179	-82.41433716				
14	42291	1	1616		16	28.03471	-82.4164	120	12/7/2014 12:54	28.01302	-82.41511536				
15	42291	1	1616		16	28.03471	-82.4164	120	12/7/2014 12:54	28.01302	-82.41511536				
16	42291	1	1616		16	28.03471	-82.4164	60	12/7/2014 12:55	28.00486	-82.41420746				
17	42275	0	6968		1	27.95476	-82.4583	180	12/7/2014 12:56	27.97809	-82.42237854				
18	42291	1	1616		16	28.03471	-82.4164	0	12/7/2014 12:57	27.9963	-82.41426086				
19	42275	0	2980		32	27.97998	-82.4223	180	12/7/2014 12:57	27.98496	-82.42241669				
20	42291	1	1616		16	28.03471	-82.4164	0	12/7/2014 12:57	27.9963	-82.41426086				
21	42275	0	2980		32	27.97998	-82.4223	180	12/7/2014 12:58	27.98883	-82.42115021				
22	42291	1	1616		16	28.03471	-82.4164	120	12/7/2014 12:58	27.99594	-82.41426086				
23	42275	0	2980		32	27.97998	-82.4223	180	12/7/2014 13:00	27.98878	-82.41429901				
24	42291	1	1645		32	27.99536	-82.4142	60	12/7/2014 13:00	27.98926	-82.41424561				
25	42275	0	2980		32	27.97998	-82.4223	180	12/7/2014 13:01	27.98878	-82.41429901				

Route 5 42341 42342 42343 42344 42345 42346 42347 42348 42349 42350 42351 42352 42353 42354 Graph 4 ... 100%

Figure 44: Dataset Results for Route 5

2. Separating data for each trip: Records for each trip are separated so that additional data can be processed (Figure 45). All of these records are sorted by timestamps.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
	trip_id	direction	stop_id	stop_seq	stop_lat	stop_lon	arrival_de	timestamp	position_l	position_longitude	Dist	Cum Dist			timestamp	
1	42359	1	6497	1	28.06642	-82.4299	60	12/16/2014 19:01	28.06553	-82.4275	0.0000	0.0000			12/16/2014 19:01	
2	42359	1	6497	1	28.06642	-82.4299	0	12/16/2014 19:01	28.06551	-82.4265	0.0600	0.0600			12/16/2014 19:01	
3	42359	1	6497	1	28.06642	-82.4299	0	12/16/2014 19:02	28.06551	-82.4265	0.0000	0.0600			12/16/2014 19:02	
4	42359	1	6497	1	28.06642	-82.4299	60	12/16/2014 19:02	28.0655	-82.4253	0.0749	0.1349			12/16/2014 19:02	
5	42359	1	6497	1	28.06642	-82.4299	60	12/16/2014 19:03	28.0655	-82.4253	0.0000	0.1349			12/16/2014 19:03	
6	42359	1	6497	1	28.06642	-82.4299	60	12/16/2014 19:03	28.06594	-82.4209	0.2651	0.4000			12/16/2014 19:03	
7	42359	1	6497	1	28.06642	-82.4299	60	12/16/2014 19:04	28.06594	-82.4209	0.0000	0.4000			12/16/2014 19:04	
8	42359	1	6497	1	28.06642	-82.4299	60	12/16/2014 19:04	28.06378	-82.4203	0.1550	0.5550			12/16/2014 19:04	
9	42359	1	6497	1	28.06642	-82.4299	60	12/16/2014 19:05	28.06378	-82.4203	0.0000	0.5550			12/16/2014 19:05	
10	42359	1	6497	1	28.06642	-82.4299	120	12/16/2014 19:05	28.06141	-82.4196	0.1677	0.7227			12/16/2014 19:05	
11	42359	1	6497	1	28.06642	-82.4299	120	12/16/2014 19:06	28.06141	-82.4196	0.0000	0.7227			12/16/2014 19:06	
12	42359	1	6497	1	28.06642	-82.4299	120	12/16/2014 19:06	28.05802	-82.419	0.2381	0.9608			12/16/2014 19:06	
13	42359	1	6497	1	28.06642	-82.4299	120	12/16/2014 19:07	28.05802	-82.419	0.0000	0.9608			12/16/2014 19:07	
14	42359	1	6497	1	28.06642	-82.4299	60	12/16/2014 19:07	28.05717	-82.4148	0.2612	1.2220			12/16/2014 19:07	
15	42359	1	6497	1	28.06642	-82.4299	60	12/16/2014 19:08	28.05717	-82.4148	0.0001	1.2221			12/16/2014 19:08	
16	42359	1	6497	1	28.06642	-82.4299	60	12/16/2014 19:08	28.05481	-82.4134	0.1839	1.4060			12/16/2014 19:08	
17	42359	1	6497	1	28.06642	-82.4299	60	12/16/2014 19:09	28.05481	-82.4134	0.0000	1.4060			12/16/2014 19:09	
18	42359	1	6497	1	28.06642	-82.4299	120	12/16/2014 19:09	28.05452	-82.416	0.1637	1.5697			12/16/2014 19:09	
19	42359	1	6497	1	28.06642	-82.4299	120	12/16/2014 19:10	28.05452	-82.416	0.0000	1.5697			12/16/2014 19:10	
20	42359	1	6497	1	28.06642	-82.4299	180	12/16/2014 19:10	28.0545	-82.416	0.0012	1.5709			12/16/2014 19:10	
21	42359	1	6497	1	28.06642	-82.4299	180	12/16/2014 19:11	28.0545	-82.416	0.0000	1.5709			12/16/2014 19:11	
22	42359	1	6497	1	28.06642	-82.4299	240	12/16/2014 19:11	28.0545	-82.416	0.0000	1.5709			12/16/2014 19:11	
23	42359	1	6497	1	28.06642	-82.4299	240	12/16/2014 19:12	28.0545	-82.416	0.0000	1.5709			12/16/2014 19:12	
24	42359	1	6497	1	28.06642	-82.4299	180	12/16/2014 19:12	28.04827	-82.4164	0.4309	2.0018			12/16/2014 19:12	
25	42359	1	6497	1	28.06642	-82.4299	180	12/16/2014 19:13	28.04414	-82.4165	0.2854	2.2872			12/16/2014 19:13	
26	42359	1	6497	1	28.06642	-82.4299	180	12/16/2014 19:13	28.04388	-82.4164	0.0185	2.3057			12/16/2014 19:13	
27	42359	1	6497	1	28.06642	-82.4299	180	12/16/2014 19:14	28.04388	-82.4164	0.0001	2.3058			12/16/2014 19:14	
28	42359	1	1616	16	28.03471	-82.4164	60	12/16/2014 19:15	28.03224	-82.4164	0.8041	3.1098			12/16/2014 19:15	
29	42359	1	1616	16	28.03471	-82.4164	60	12/16/2014 19:15	28.02974	-82.4164	0.1733	3.2832			12/16/2014 19:15	
30	42359	1	1616	16	28.03471	-82.4164	60	12/16/2014 19:16	28.02974	-82.4164	0.0000	3.2832			12/16/2014 19:16	
31	42359	1	1616	16	28.03471	-82.4164	0	12/16/2014 19:16	28.02266	-82.4157	0.4915	3.7747			12/16/2014 19:16	
32	42359	1	1616	16	28.03471	-82.4164	0	12/16/2014 19:17	28.02266	-82.4157	0.0000	3.7747			12/16/2014 19:17	
33	42359	1	1616	16	28.03471	-82.4164	0	12/16/2014 19:17	28.01843	-82.4144	0.3023	4.0769			12/16/2014 19:17	
34	42359	1	1616	16	28.03471	-82.4164	0	12/16/2014 19:18	28.01843	-82.4144	0.0000	4.0769			12/16/2014 19:18	
35	42359	1	1616	16	28.03471	-82.4164	60	12/16/2014 19:18	28.01839	-82.4144	0.0022	4.0792			12/16/2014 19:18	
36	42359	1	1616	16	28.03471	-82.4164	60	12/16/2014 19:19	28.01839	-82.4144	0.0000	4.0792			12/16/2014 19:19	
37	42359	1	1616	16	28.03471	-82.4164	60	12/16/2014 19:19	28.01192	-82.4153	0.4508	4.5300			12/16/2014 19:19	
38	42359	1	1616	16	28.03471	-82.4164	60	12/16/2014 19:20	28.01192	-82.4153	0.0000	4.5300			12/16/2014 19:20	
39	42359	1	1616	16	28.03471	-82.4164	0	12/16/2014 19:20	28.00612	-82.4142	0.4065	4.9366			12/16/2014 19:20	
40	42359	1	1616	16	28.03471	-82.4164	0	12/16/2014 19:21	28.00612	-82.4142	0.0000	4.9366			12/16/2014 19:21	
41	42359	1	1616	16	28.03471	-82.4164	0	12/16/2014 19:21	28.00227	-82.4142	0.2661	5.2027			12/16/2014 19:21	
42	42359	1	1616	16	28.03471	-82.4164	0	12/16/2014 19:22	28.00227	-82.4142	0.0001	5.2028			12/16/2014 19:22	
43	42359	1	1616	16	28.03471	-82.4164	-60	12/16/2014 19:22	27.99632	-82.4142	0.4111	5.6139			12/16/2014 19:22	
44	42359	1	1616	16	28.03471	-82.4164	-60	12/16/2014 19:23	27.99632	-82.4142	0.0000	5.6139			12/16/2014 19:23	
45	42359	1	1616	16	28.03471	-82.4164	-60	12/16/2014 19:23	27.99632	-82.4142						

Figure 45: Dataset for Individual Trips

3. Calculating the distance between latitude/longitude points. An Excel equation was used, as follows:

$$\text{Distance} = 6371 * \text{ACOS}(\text{COS}(\text{RADIANS}(90 - I2)) * \text{COS}(\text{RADIANS}(90 - I3)) + \text{SIN}(\text{RADIANS}(90 - I2)) * \text{SIN}(\text{RADIANS}(90 - I3)) * \text{COS}(\text{RADIANS}(J2 - J3))) / 1.609$$

Where I2 = Latitude 1 in decimals

J2 = Longitude 1 in decimals

I3 = Latitude 2 in decimals

J3 = Longitude 2 in decimals

The unit for this calculation is a mile.

The distance between two points is calculated using the previous formula. Then, to obtain the total distance covered by a bus, the cumulative distance is calculated by adding the previous distances

(Figure 46).

Figure 46: Distance and Cumulative Distance

4. After calculating the distance, the timestamp versus distance graphs for each trip is prepared (Figure 47).

Figure 47: Timestamp and Distance Data

Graphs were created for a specific weekday; in this case, a Tuesday, for three consecutive weeks: December 9, 2014, December 16, 2014 and December 23, 2014. The purpose was to observe potential variations (e.g., headways, times, distances) from one week to another.

Time-space diagrams can help visualize bus bunching. Figures 48, 49, and 50 show southbound trips on HART's Route 5 for Tuesday, for three consecutive weeks, starting on December 9, 2014. The goal was to find a pattern of changes from week to week on a particular day. As can be seen in the abovementioned figures, with the expectation of minor deviations, the trips had uniform headways, which makes for a good case and ideal situation.

Figure 48: Hart Transit Route 5

Figure 49: Hart Transit Route 5

Figure 50: Hart Transit Route 5

In addition, plots are created with the use of the trip latitude and longitude data. Figure 51 shows the location of the vehicles on a directional trip along Route 5. As expected, the vehicles follow the same path as the Route 5 alignment presented in Figure 51. This plot was created using archived GTFS-realtime data, which is useful for visualizing bus locations on a map. As can be seen, this is a rather uniform set of location points. For a specific view of a particular location or a segment of the route, maps can be zoomed in and out. Overall, the plot can be used as a planning tool, which can help determine segments that may need special attention.

Figure 51: HART Transit Route 5 (Trip #42359 Southbound)

In the case of MDT Route 288, a different strategy was used: bidirectional trips for a weekday were selected during 5:30 a.m. and 10:00 a.m. for a particular day (October 15, 2015). By using the space-time diagram (Figure 52), areas of concern are identified and analyzed. The results show the bus bunching and gap issues, with the more prominent cases in the eastbound direction. It appears that for this particular day, the eastbound trips between 7:00 a.m. and 9:00 a.m. start having problems at SW 88 Street/SW 127 Avenue. This needs to be investigated further, as it could be a repeating issue or an isolated issue for that particular day due to a traffic incident or other unexpected event. If this is a repeated issue, transit operators need to monitor and identify the reasons for this occurrence at this location and find the most appropriate solution. The causes may vary, such as scheduling issues that can be resolved by modifying the timetable, or a traffic light problem that can be solved by better signal coordination. Nevertheless, addressing this issue and working with the appropriate transit department or organization can improve the reliability of the route.

Figure 52: Miami-Dade Transit Route 288

Figure 53 depicts the geographic locations of a vehicle on an eastbound trip (Trip 3502699) along Route 288 eastbound. Areas where gaps occur can be determined by using the archived data of vehicle locations. It appears that the locations west of SW 122 Avenue are uniform, while the locations to the east are more spread out. The reasons for this effect could vary and have different explanations, such as traffic issues, spacing of stops, stop ridership, bus speed, etc. Again, knowing the traffic and road conditions of a route can improve the reliability of the route. Furthermore, maps are a great tool for monitoring transit vehicles and identifying potential bunching issues so that appropriate action can be taken for prevention of such occurrences. One approach is to safely

speed up vehicles while slowing down others. Holding buses at timepoints may work, but this has a disadvantageous effect on the passengers waiting inside the vehicle. Modifying the vehicle speeds to control bunching has a less negative impact.

Figure 53: Miami-Dade Transit Route 288 (Trip #3502699 Eastbound)

Figure 54 displays another way to visualize trips along a particular stop on Route 288. This time diagram shows different eastbound trips on Route 288, at the SW 88 Street/SW 127 Avenue timepoint between 6:00 a.m. and 8:00 a.m. This figure also shows some potential bunching around 6:00 a.m., bunching at 7:00 a.m., and somewhat normal service at around 8:00 a.m.

SW 88 St/SW 127 Ave

Figure 54: Trips by Time of Day on SW 88 Street/SW 127 Avenue

PRODUCING ON-TIME PERFORMANCE FROM GTFS- REALTIME DATA

The GTFS and GTFS-realtime data specifications have revolutionized the transit industry – for the first time, data from a large number of agencies are available in the same format. While GTFS and GTFS-realtime were developed to power customer-facing transit apps, a unified format also makes data analysis and processing across multiple agencies much simpler. Data no longer needs to be converted by hand from a proprietary data format (usually specific to a vendor) into a common format for analysis. Additionally, having the data openly available in a common format straight from the data source (i.e., transit agency) also allows the development of software tools to automate the data collection and analysis process for *any* agency that openly shares their GTFS and GTFS-realtime data.

This section proposes a process and proof-of-concept software tool for generating schedule deviation information directly from GTFS and GTFS-realtime data, which can then be used to determine on-time performance (OTP).

GTFS-realtime – The Raw Data

As presented in the “Understanding GTFS-realtime” section, there are two primary types of real-time information shared in a GTFS-realtime feed that are directly relevant to calculating OTP – Vehicle Positions and Trip Updates.

Vehicle Positions

A GTFS-realtime Vehicle Positions feed contains the latitude and longitude of the vehicle, along with other information such as the trip ID the vehicle is currently running. Here is an example snapshot of a Vehicle Position from HART’s feed for a single vehicle – a GTFS-realtime feed will include this data for all vehicles currently running routes:

```
entity {
  id: "vehicle_position_1012"
  vehicle {
    trip {
      trip_id: "183242"
    }
    position {
      latitude: 28.069168
      longitude: -82.45089
      bearing: 75.0
      speed: 12.0
    }
    vehicle {
      id: "1012"
    }
  }
}
```

Trip Updates

A GTFS-realtime Trip Updates feed contains the predictions for when a vehicle will arrive and depart at upcoming stops. Here is an example snapshot of a Trip Update from HART’s feed for a single prediction – a GTFS-realtime feed will typically include this data for all trips in blocks that currently have vehicles running routes:

```
entity {
  id: "trip_update_183242"
  trip_update {
    trip {
      trip_id: "183242"
    }
    stop_time_update {
      stop_sequence: 57
      arrival {
        delay: 120
      }
      departure {
        delay: 120
      }
      stop_id: "4027"
    }
    vehicle {
      id: "1012"
    }
  }
}
```

The “delay” value is the estimated number of seconds that this vehicle is running early (a negative value) or late (a positive value). In the above example, the vehicle is predicted to be running 2 minutes behind schedule for its arrival and departure at stop ID 4027 as part of trip ID 183242. Some advanced AVL systems provide multiple stop_time_updates per trip, meaning that the system predicts arrivals for more than one upcoming stop in the current trip. These systems presumably use real-time or historical information when generating per-stop predictions, instead of just generating a single prediction for the vehicle.

Producing Schedule Deviation from GTFS-realtime Data

OTP is calculated using schedule deviation measurements. Schedule deviation is the difference between when a vehicle is scheduled to arrive or depart at a stop or timepoint, and when it actually arrives or departs. As mentioned earlier in this report, in the case that arrival and departure deviations are known, departure deviations are preferred for measuring on-time performance (a vehicle could arrive early at a stop, but depart on-time, not adversely affecting the rider).

At a glance, it may seem like that schedule deviation information is readily available from the Trip Updates feed – there is a “delay” value contained within the data itself. However, one must

remember that GTFS-realtime was designed primarily for user-facing applications, such as telling a rider how long it will be until their bus arrives. As a result, the “delay” value is a *prediction* of a future event (i.e., the bus arriving/departing at a stop it has not yet visited), which may not be the same as when the vehicle actually arrived/departed at that stop. Other Trip Update fields such as stop_id and stop_sequence may help in determining when a vehicle passed a stop (and as a result the schedule deviation) – for example, when the stop_id and stop_sequence changes, one could assume that the vehicle just passed the previous stop_id and is on its way to the next stop. However, the GTFS-realtime specification does not require this behavior – a feed could still be publishing a stop_id with a prediction after a vehicle passes that stop. The GTFS-realtime specification simply says that consumers should simply propagate the delay information to all remaining stops in the trip. As a result, Trip Updates alone are not a good candidate for calculating schedule deviation.

A better approach is to leverage the Vehicle Position format and the location information contained in that feed to determine when a vehicle arrived/departed at a stop.

Determining When a Vehicle Arrived at a Stop

Figure 55 shows example GPS data from a transit vehicle (red connected dots), along with bus stops for a transit route, it is currently running (blue connected dots).

Figure 55: Example GPS Data and Closest Stops for a Vehicle Running a Transit Route

Each GPS location has a timestamp associated with it. AVL systems typically send GPS updates from vehicles at a fixed time interval (e.g., every 30 seconds), and/or when a vehicle passes a particular geographic location (i.e., a timepoint). Timepoints may be co-located with bus stop locations, but they can also be assigned to arbitrary geographic locations if allowed by the CAD/AVL system. Transit agencies typically use schedule deviation at timepoints to measure OTP, and there are typically far fewer timepoints than bus stops.

The bus stops locations are defined in the GTFS stops.txt file, and the order of which the stops are visited for a particular trip (A-B-C-D in Figure 55) is defined in GTFS stop_times.txt. stop_times.txt also contains an optional field “timepoint” where agencies can indicate if a stop is a timepoint and strictly adhered to by the transit vehicle (1), or if the scheduled arrival/departure time at that stop is simply an approximation (0).

By measuring the distance between the GPS positions and stops, the closest bus stops to each GPS point can be determined. The closest stop for each GPS location is shown using the dashed blue line in Figure 55 and example distance values to illustrate this concept are shown in Table 24.

Table 24: Calculating Simplified Schedule Deviation from GPS Data

GPS Point	Closest Stop	Distance to Stop (meters)	Scheduled Arrival	GPS Timestamp	Simplified Schedule Deviation (seconds)
1	A	30 m	9:00:00am	9:01:30am	90 sec
2	B	60 m	9:05:00am	9:02:30am	-150 sec
3	B	20 m	9:05:00am	9:05:30am	30 sec
4	C	70 m	9:10:00am	9:08:00am	120 sec
5	D	50 m	9:15:00am	9:13:30am	90 sec

A simplified schedule deviation can be calculated by subtracting the GPS timestamp from the scheduled arrival/departure time from that stop – example scheduled arrivals and GPS timestamps are shown in Table 24, with the far right column holding the simplified schedule deviation.

The primary challenge in calculating schedule deviation is determining exactly when a vehicle arrived or departed at a stop/timepoint. As can be seen in Figure 55, GPS locations will never fall exactly on the bus stop location due to the offset of the bus stop location from the road, the update rate of the GPS, and error in the GPS positions and geocoded bus stop positions. As a result, the simplified schedule deviation shown in Table 24 approximates the true schedule deviation. This approximation will be more accurate in the cases where the GPS update is closer to the stop – for example, Stop B has two GPS updates associated with it. The second update, GPS Point 3, is closer than the first (GPS Point 2). Therefore, the simplified schedule deviation calculated for Stop B using GPS Point 3 (30 seconds) is likely to be closer to the true deviation than GPS Point 2 (-150 seconds).

The accuracy of the schedule deviation approximation can have an impact on calculating OTP. For example, using the closer GPS Point 3 for Stop B resulted in a schedule deviation of 30 seconds (running late). If the further GPS Point 2 was used that appeared before the bus stop, the schedule deviation would have been 150 seconds (running early). Less frequent GPS updates would yield larger potential error in approximations.

As mentioned above, OTP is typically calculated using schedule deviation at timepoints, which may be co-located with bus stops. Therefore, some of the schedule deviations for stops shown Table 24 may not be used in calculating OTP. For example, if the only timepoints on this route were Stops A and C, average schedule deviation for this data would be:

$$\text{avg. schedule deviation} = (90 + 120) / 2 = 105 \text{ seconds}$$

However, if Stops A, B, C, and D are all timepoints, then the average schedule deviation would be:

$$\text{avg. schedule deviation} = (90 + 30 + 120 + 90) / 4 = 82.5 \text{ seconds}$$

Therefore, it is important that agencies provide the GTFS stop_times.txt timepoint field so it can be used to accurately calculate schedule deviation and OTP.

The “Conclusions, Observations, and Next Steps” chapter outlines future work related to producing better approximations of schedule deviation.

Proof-of-Concept Open-Source Software Tool

To demonstrate the calculation of schedule deviation and OTP from GTFS and GTFS-realtime data, the research team created the “On-Time Performance Calculator” open-source project, which is available on Github at <https://github.com/CUTR-at-USF/ontime-performance-calculator> (Figure 56).

Figure 56: The On-Time Performance Calculator Open-source Tool

This tool uses two datasets as input:

1. Archived GTFS-realtime data – The open-source project GTFSrDB (<https://github.com/mattwigway/gtfsrdb>) can be used to archive GTFS-realtime data into a relational database such as Microsoft SQL Server.
2. GTFS data - A zip file containing GTFS data from the same time period as the archived GTFS-realtime data.

When executed, the tool loads the data and calculates the simplified schedule deviation for each GPS point and the closest stop as shown in Figure 55 and Table 24, using the following algorithm:

1. Find the closest stop S in Trip A to a GPS point for Trip A. Then, find the scheduled time of closest stop S and whether or not it is a timepoint from the GTFS data. At this point, we also have the timestamp for P from the GTFS-realtime database.
2. Calculate the simplified schedule deviation (timestamp P – scheduled time at stop S) and store it as a field in the GTFS-realtime vehicle positions database table.
3. If there are multiple GPS points assigned to the same stop S for the same Trip A, update the database via a SQL query to mark, which is the closest point to that stop S for each day of service – the schedule deviation for this point is what should be used to calculate OTP.

Additional documentation for this tool is located in the README page of the project on Github.

Future work can expand the capabilities of this tool to calculate schedule deviation using a number of heuristics, as well as OTP, as discussed in the “Conclusions, Observations, and Next Steps” chapter. The tool could also be improved to use the capabilities of spatial databases to reduce execution time when determining proximity of GPS data to stops.

CONCLUSIONS, OBSERVATIONS, AND NEXT STEPS

Automatic Vehicle Location (AVL) systems are computer-based vehicle tracking systems that measure the real-time positions of vehicles, which can relay the data back to a central location. AVL systems are most frequently used when identifying the location of vehicles for a variety of purposes including fleet management. AVL systems track vehicles using location technologies, but GPS is the most effective and accurate tracking system for transit. The GPS system works through a network of orbiting satellites that transmit signals to the ground at polling intervals. Special receivers on each vehicle read the available signals and triangulate to determine their position. The geographic location, along with the date, time, and other operational data is sent to the transit agency to be used immediately for daily operations such as scheduling and service planning. Furthermore, the use of historical AVL data can help analysts identify recurring problems that occurred in the past, which can help develop solutions to these problems.

AVL data can be used to measure, monitor, and improve service reliability, also referred to as on-time performance (OTP). On-time performance is very important to transit customers. Strategies can improve customer satisfaction and attract new transit riders. Transit agencies understand that reliability is necessary in order to improve service and this can be achieved at a relatively low cost. Therefore, the benefit-to-cost ratio of improving on-time performance is expected to be significant.

To improve transit service reliability, there is a need for a systematic review of archived AVL data and the use of real-time data. This can help monitor the transit system and identify recurring service problems, as well as find conditions in the data that exist in the time period preceding the service problem.

Initial studies regarding the analysis of service reliability mainly concentrate on configuring AVL systems and the potential use of data to analyze transit performance. Later studies focus on developing methodologies for analyzing service reliability and strategies to restore services. There are several methodologies derived from the proposed indicators by TCRP Report 165. Unfortunately, most of these analyses are limited to historical archived data, with the exception of a few studies; the findings are more useful for planning rather than operations. Early identification of these conditions can help transit agencies make intelligent decisions not only from the planning perspective, but also for the real-time operations, which can determine the best course of action for avoiding service degradation. This can also enhance the quality of service and customer satisfaction.

According to the literature review, there are several weaknesses in the performance measurement of transit services. Quality of transit service can degrade for many reasons, both externally and internally. Reasons for the degradation range from inadequate service network design, mismanagement of resources (vehicle, crew), and very high passenger load, to more stochastic causes, such as weather, driver behavior, policy, etc. However, previous works reference schedule deviation and variability of trip running time as the main causes for service degradation.

AVL systems play an important role in overcoming most of the limitations discussed above, whether related to service performance indicators or service improvements or restoration. Most agencies use the AVL system for service monitoring, but only a few agencies take advantage of the immense amount of data and information generated by the system. Furthermore, the systems currently used for service monitoring or bus dispatching are limited in the area of detecting early indications of possible service failure. This study attempts to address these issues and presents suggestions for improving service reliability. Again, the ideas presented can be used on a limited scale. A better alternative is the development of a software tool that takes full advantage of the archived and real-time AVL data for improving transit service reliability, as highlighted throughout this report.

In order to improve service reliability, different elements were investigated from the transit agency point of view, such as improving on-time performance and monitoring schedule adherence and vehicle bunching to prevent service degradation. Static and real-time GTFS data from HART and data from the MDT AVL system are used. The HART datasets included both static and real-time data. To help prepare the GTFS data for this project's data analysis, a diagram was developed by the research team, which is shown in Figure 10. It depicts the GTFS static and real-time data elements that were used, as well as its relationships, and the expected outputs.

The two MDT datasets included archived data at timepoint level and data compiled from real-time outputs. The goal was to demonstrate how service reliability is improved by modifying the timetables and by monitoring schedule adherence and bus bunching, which is accomplished by using transit planning, scheduling, and operations strategies and techniques.

For on-time performance, several data issues are discussed, such as the quality of the data, schedule adherence parameters, missing records, extreme values or outliers, end-of-line timepoints, and other issues that may have an impact on the way OTP is calculated, measured, monitored, and managed. Service degradation is addressed through the monitoring of bus schedule adherence and of conditions such as the fluctuating spacing among vehicles, which can lead to bus bunching.

The importance of creating a tool or tools to help transit agencies improve service reliability is emphasized throughout this report. Figure 20 presents a diagram of a potential reliability system. In this architectural view, there are two components of the system: On-Time Performance and Bus Bunching. In the first component, there are three sub-components: early, on time, and late. Feedback is sent to the system for any responses (early, on time, and late).

The alarms could be set up in stages based on the severity and the level of detail (e.g., trip, route, or system level). For instance, for on-time performance, an alarm for Stage 3 could be set to be lower than 50% for a trip. This can alert an operations supervisor to take immediate action before the service deteriorates any further. Ideally, the issue should be addressed at Stage 1 when OTP is set up to be something like 70%. Similarly, bus bunching is addressed based on the headway between vehicles. That is, if the actual headway is half of the scheduled headway, the Stage 2

alarm is triggered. In any case, the settings should be based on agency policy and availability of resources.

Results from Case Studies

The examples presented in the Case Studies section can be used as a reference for strategies and techniques that improve service reliability. The following are various on-time performance parameters that are used by transit agencies. Buses are considered on time when they are within the following parameters:

- a. 5 minutes late and 2 minutes early [-5, 2].
- b. 5 minutes late and 1 minutes early [-5, 1].
- c. 5 minutes late and 0 minutes early [-5, 0].

Different statistical distributions were considered in the calculations, such as Normal, Three-Parameter Lognormal, Smallest Extreme Value, and Largest Extreme Value. Of all of these distributions, the Smallest Extreme Value distribution performed the best for the datasets used in this project. The data analysis was conducted at the Route, Direction, Trip, and Timepoint levels.

The use of different examples, strategies and techniques were used to demonstrate how on-time performance can be improved. For instance, in the case of Trip 3491108, a series of steps for the improvement of on-time performance are shown below by 1) adjusting the times in the timetable (7% improvement); 2) controlling the variability (44% improvement), and 3) shifting the times in the timetable (2.71% improvement). The basic optimization steps are as follows:

- 1) Improve on-time performance by adjusting the times at the timepoints.
- 2) Reduce the schedule adherence variability by real-time monitoring and control.
- 3) Shift the times in timetables to maximize OTP.

In public transportation, bus bunching is a condition that occurs when buses running in the same direction get so close to one another that they travel in a bunch or group along the same route. This occurs when at least one of the vehicles is unable to maintain its schedule and ends up drawing closer to the other vehicles. Bus bunching results in unreliable service, long wait times for passengers, overcrowded vehicles, and near empty buses.

Prevention of Bus Bunching

Bus bunching can be prevented in several ways. Following are some commonly used strategies to prevent bunching.

1. Hold buses at certain timepoints along the route.
2. Allow crowded buses to skip certain stops.

3. Maintain a maximum number of passengers on the bus.

Calculation of Bus Bunching

Our approach for calculating bus bunching is based on time-space diagrams of the trips along a route. For the data analysis, the HART General Transit Feed Specification (GTFS) data was stored in an SQL Server database, and Route 5 was used for the analysis and creation of the figures. The HART datasets included both static and real-time GTFS data.

The basic steps needed to accomplish this task include the following:

1. Collect data for a route
2. Separate data for each trip
3. Calculate the distance
4. Create time-space graphs

From a planning perspective, time-space diagrams can help visualize bus bunching. That is, when consecutive trips grow further apart, create gaps or start bus bunching, negative impacts occur and affect the quality of service. Figures 48, 49, and 50, which were generated from GTFS data, show southbound trips on HART's Route 5 on a specific day; in this case, a Tuesday, during three consecutive weeks starting on December 9, 2014. The purpose for this task was to observe any changes from week to week on a particular day. As can be seen, with the exception of minor deviations, the trips had uniform headways.

In addition, plots are created using the trip latitude and longitude data. Figure 51 shows the location of the vehicles on a directional trip along Route 5. As expected, the vehicles follow the same path as the Route 5 alignment presented in Figure 23. This plot was created using archived real-time GTFS data, which is useful for visualizing bus locations on a map. This is a rather uniform set of location points. Maps can be zoomed in and out for a specific view of a particular location or a segment of the route. Overall, plots are used as planning tools, which can help determine segments that may need special attention.

From the transit operator's perspective, monitoring the locations of vehicles real-time on a map or other graphical representation of the vehicle locations can help maintain the headways. If buses appear to be getting too close to each other, some action may need to be taken.

Again, knowing the traffic and road conditions of a route can improve the reliability of the route. Furthermore, maps are a great tool for monitoring transit vehicles and identifying potential bunching issues so that appropriate action can be taken for prevention of such issues. One approach is to safely speed up vehicles while slowing down others. Holding buses at timepoints may work, but it has a disadvantageous effect on the passengers waiting inside the vehicle. Modifying the vehicle speeds to control bunching has a less negative impact.

Due to the complexity of using GTFS and AVL data, the ideal solution is to develop a computerized tool or tools to help transit agencies provide reliable services. With a computerized system, management and staff can easily monitor, manage, and improve the reliability of the transit system.

General Observations

Transit reliability is a desired outcome and is probably one of the least expensive services a transit agency can perform to improve transit service. This research concentrates on two techniques that improve the reliability of a transit service: on-time performance and bus bunching. These techniques are linked mainly to transit scheduling and operation. Nevertheless, planning can also play a role in analyzing data to assist the scheduling and operations departments.

Data Issues

When using AVL data for on-time performance, there are some issues that need to be taken into consideration, such as cleaning the data, which may require deletion of unnecessary fields, end of the line timepoints, and outliers that use statistical methods like the Box Plot method. Similarly, for bunching detection, only the necessary fields should be included, such as geographic location and timestamp. In addition, it should be pointed out that manipulating large amounts of data and complex queries can be time-consuming, which needs to be taken into consideration if the agency has limited resources.

Another issue is using the mean or median schedule adherence values from AVL systems. It is advisable not to use these values to adjust the times in the timetables, as they may cause buses to arrive and depart early. By using mean or median values from AVL systems, there is a good probability that buses arrive early and decrease on-time performance, which is not a desired outcome. In order to optimize on-time performance, buses should arrive/depart within the on-time performance parameters. For illustration purposes, and assuming the use of normal distribution, consider that the middle of the OTP parameters for $[-5, 2]$ is -1.5 ; for $[-5, 1]$ is -2 , and for $[-5, 0]$ is -2.5 . Using the $[-5, 0]$ case to optimize OTP, the mean value of the schedule adherence values should be set to be 2.5 minutes late (not the mean = 0, as commonly used). As a result, buses would have to depart 2.5 minutes later on average. In the case of $[-5, 2]$, this value is only 1.5 minutes late, which is closer to the time that passengers expect the buses to depart.

To avoid conflicts between two different goals (maximizing OTP and adhering to schedule), it is suggested that the $[-5, 2]$ parameters be used. Ideally, the late and early on-time performance parameters should be equal.

While GTFS-realtime data offers huge potential that would allow the automated processing of data from a large number of agencies in a standardized format, GTFS-realtime does not directly provide schedule deviation or OTP measurements. As a result, schedule deviation must be calculated using

vehicle position information in the GTFS-realtime feed and the bus stop locations in the GTFS data. OTP can then be calculated from the schedule deviations. As discussed in the chapter “Producing On-Time Performance from GTFS-realtime Data,” these schedule deviations are approximations, the accuracy of which is affected by the vehicle GPS update rate, bus stop location and vehicle GPS errors, and offset of the bus stop location from the road. For example, newer AVL systems that provide GPS updates for each vehicle every 30 seconds would typically result in more accurate schedule deviation values than an older AVL system that updates every 2 minutes.

Future research should focus on better understanding how different approximations of schedule deviations can affect OTP. Possible heuristics for producing better approximations of schedule deviation from bus stop location and GPS data include:

1. **Distance-based estimation using straight lines** – The true arrival/departure time can be estimated by using the straight-line distance from the GPS location to the stop to estimate how long it would take the bus to travel this distance.
2. **Distance-based estimation using road network information** – The true arrival/departure time can be estimated by using the distance traveled on the road network from the GPS location to the stop to estimate how long it would take the bus to travel this distance on the road. This could use either the GTFS shapes.txt data if provided, or a separate road network database.
3. **History-based interpolation or regression using archived vehicle location data** – The true arrival/departure time at the stop can be interpolated or calculating using regression by using historical travel time information based on archived vehicle location data on that same route.
4. **Machine-learning-based estimation using archived vehicle location data** – The true arrival/departure time can be estimated using machine learning techniques and a large history of vehicle position information.
5. **Arrival prediction-based estimation** – The true arrival/departure time can be estimated by using the Trip Updates feed data (i.e., predicted vehicle arrivals) along with the vehicle GPS location data.
6. **Hybrid estimation** – Several of the above techniques could be combined to improve the accuracy of the approximations.

It should be noted that calculating schedule deviations for determining OTP is conceptually very similar to predicting the arrival and departure times of the transit vehicle in real-time for passenger information systems – the goal of both processes is to estimate when a vehicle arrives or departs at a stop given vehicle position information. Therefore, the literature for prediction techniques, including regression (Biagioni et al. (2011), Core. (2016), Zimmerman et al. (2011)) and machine learning algorithms (Bin (2006), Ding (2000), Chien et al. (2002), Chien et al. (2003), Gurmur (2014), Jeong et al. (2004), Jeong et al. (2005), Wang (2014), Yu et al. (2011)), should be examined in the context of schedule deviation and OTP.

Future work should also examine how the selection of timepoint locations (including co-located vs. not co-located with bus stops) can affect the calculation of on-time performance.

As mentioned earlier, when calculating OTP departure schedule deviations are preferred to arrival schedule deviations. The accuracy of calculating actual arrives vs. actual departures should be examined to see if one is more accurate than the other. For example, if vehicles tend to bias towards reporting a GPS location when approaching a stop rather than reporting a position when leaving that stop, then arrival schedule deviations could be more accurate than departure schedule deviations.

There are several caveats that must be considered when calculating schedule deviation and on-time performance from GTFS and GTFS-realtime data, as outlined below.

1. **GTFS stops_times.txt “timepoint” field is optional** – The “timepoint” field is relatively new in GTFS, as it was officially added to the specification in February 2016. As a result, it is not yet widely adopted. Additionally, while prior to the addition of this field agencies were supposed to only include timepoints in stop_times.txt, many agencies also included interpolated scheduled times for non-timepoints. Therefore, one must exercise caution when processing datasets without the “timepoint” field and determine whether or not the data in stop_times.txt represents only timepoints, or all stops for that trip.
2. **GTFS timepoints must be co-located with stops** – Each entry in GTFS stop_times.txt, the file in which the timepoint field is defined, must include a stop ID. As a result, timepoints can only be created at stop locations in a GTFS dataset. Future work could focus on improving the specification to allow specifying arbitrary locations for timepoints – for example, additional fields to signify a timepoint could be added to GTFS shapes.txt.
3. **Timestamp of the vehicle position vs. timestamp of the feed update** – In GTFS-realtime, there are two different timestamp fields that are both optional. The FeedHeader element includes a timestamp that represents the most recent time that the contents of the feed were updated – this element is typically populated in feeds, as it can simply be the server time when the data was last published. The VehiclePosition element includes a timestamp that represents the time at which the vehicle position was calculated (e.g., GPS timestamp) – this is the field needed to accurately calculate schedule deviation.
4. **Duplicate data generated when polling GTFS-realtime feed** – When continuously fetch the real-time data and inserting it into a database, tools such as GTFSrDB typically do not check if the exact same data has already been collected. Therefore, to avoid processing more than one record for the same data, one must either ensure that the software collecting data is checking the feed to see if a new version exists for that feed (i.e., if the FeedHeader timestamp has changed, and the timestamps for individual vehicle positions has changed), or filter the resulting data in the database for duplicate records.

More and more, transit agencies face conflicting goals like increasing ridership or improving reliability while reducing operational costs or having a restrictive budget. In some instances,

schedulers may have to compromise the service provided due to budget constraints. Therefore, running times and/or recovery times may need to be reduced. This can have a negative impact on on-time performance. Ideally, a good compromise could be reached, but it requires careful analysis to maintain or improve on-time performance with budget constraints.

Departmental Considerations

Scheduling – The Scheduling department plays an important role in improving on-time performance. Using AVL schedule adherence data can assist in finding the real departure times of buses at timepoints. The on-time performance parameters need to be taken into consideration to help the agency improve on-time performance. As a rule of thumb, buses departing 1 or 2 minutes later are better for OTP optimization.

Operations – The Operations department should monitor schedule adherence and monitor bus bunching. Staff should ensure that buses depart within the on-time performance parameters to improve OTP. Appropriate control measures for the improvement of service reliability include slowing down, moving faster, turning around, drop-off only, etc. The impact of good operations control is critical to providing a reliable service.

Planning – The Planning department can also assist in this effort by analyzing the data to identify areas of potential improvements. They can work with Scheduling and Operations to modify the timetables and prepare operational strategies that can be used to improve monitoring, with the goal of improving on-time performance.

Next Steps

The next steps for furthering the transit reliability knowledge include a potential list of topics identified in this research, as follows:

- Standardization of on-time performance parameters
- OTP from the users versus the transit agency perspective
- Development of a software tool to improve service reliability
- The role of Planning, Scheduling, and Operations in improving on-time performance

REFERENCES

1. Abkowitz, M., H. Slavin, R. Waksman, L. Englisher and N. Wilson. 1978. Transit Service Reliability. Report No. UMTA-MA-06-0049-78-1. U.S. Department of Transportation.
2. Abkowitz, M., and I. Engelstein. 1983. Factors Affecting Running Time on Transit Routes. *Transportation Research Part A* 17(2): 107-113.
3. Alvaro Rocha, Anna Maria Correia, Felix B. Tan, Karl A. 2014. Stroetmann, New Perspectives in Information Systems and Technologies, Volume 2, Volume 276 of Advances in Intelligent Systems and Computing. GTFS ER diagram. Springer Science & Business Media.
4. Arend, Dominic N. 1993. Choices (Version 4.0) [Computer software]. Champaign, IL: U.S. Army Corps of Engineers Research Laboratory. (CERL Report No.CH7-22510).
5. Bartholdi, III and D. Eisenstein. 2012. A Self-Coordinating Bus Route to Resist Bus Bunching, *Transportation Research Part B: Methodological*. Vol. 46, no. 4, pp. 481– 491.
6. Biagioni, J., Gerlich, T., Merrifield, T., & Eriksson, J., 2011. “Easytracker: automatic transit tracking, mapping, and arrival time prediction using smartphones.” In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems* (pp. 68-81).
7. Bin, Y., Zhongzhen, Y., and Baozhen, Y. 2006. Bus Arrival Time Prediction Using Support Vector Machines, *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, 10:4, 151-158.
8. Camus, Roberto, Longo, Giovanni, Macorini, Cristina. 2005. Estimation of Transit Reliability Level-of-Service Based on Automatic Vehicle Location Data. *Transportation Research Record: Journal of the Transportation Research Board*, No. 1927, Transportation Research Board of the National Academies, Washington, D.C. pp 277-286.
9. Carrasco Nelson. 2012. Quantifying Reliability of Transit Service in Zurich, Switzerland Case Study of Bus Line 31, *Transportation Research Record: Journal of the Transportation Research Board*, No. 2274, Transportation Research Board of the National Academies, Washington, D.C., pp. 114–125.
10. Casello, J. M., A. Nour, and B. Helliga. 2009. Quantifying the Impacts of Transit Reliability on User Cost. *Transportation Research Board 88th Annual Meeting Compendium of Papers*. Washington, DC.
11. Ceder, A. (2007). *Public Transit Planning and Operation: Theory, Modeling and Practice*. OElsevier, Oxford, UK.

12. Cevallos F. and X. Wang. 2008. ADAMS: A Data Archiving and Mining System for Transit Service Improvements. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 2063, National Research Council, Washington, D.C.
13. Cevallos, F., K. Kirwin, and R. Pearsall. 2008. "Using CAD/AVL Data for Performance Management," *Proceedings of the 10th International Conference on Applications of Advanced Technologies in Transportation*, ASCE, Athens, Greece.
14. Cevallos, F., X. Wang, Z.M. Chen, and A. Gan, January 2010. "Improving Transit On-Time Performance with AVL Data: A Timetable Approach," Presented at the 89th TRB Annual Meeting.
15. Cevallos F. and X. Wang, October 25-29, 2010. "Monitoring Transit On-Time Performance Real Time Using a Web-Service Dataset", *Proceedings of the 17th ITS World Congress*, Busan, Korea.
16. Cevallos F., X. Wang, Z. Chen, and A. Gan. 2011. Using AVL Data for Improving Transit On-Time Performance, *Journal of Public Transportation*, Vol. 14, No. 3.
17. Cham, Laura Cecilia. 2006. Understanding Bus Service Reliability: A Practical Framework using AVL /APC Data. Master Thesis, MIT.
18. Chen-Fu Liao, 2011. Data-Driven Support Tools for Transit Data Analysis, Scheduling and Planning, Intelligent Transportation Systems Institute, Center for Transportation Studies, University of Minnesota.
19. Chen, L. Yu, Y. Zhang, and J. Guo. 2009. Analyzing Urban Bus Service Reliability at the Stop, Route, and Network Levels," *Transportation Research Part A: Policy and Practice*, Vol. 43, no. 8, pp. 722–734.
20. Chen, Q., Adida, E., Lin, J. 2013. Implementation of an Iterative Headway-Based Bus Holding Strategy with Real-time Information.
21. Chien, S., Ding, Y., and Wei, C. 2002. "Dynamic Bus Arrival Time Prediction with Artificial Neural Networks", *ASCE Journal of Transportation Engineering*, 128(5), p. 429-438.
22. Chien, S., Chen, M., and Liu, X. 2003. "Use of Neural Network / Dynamic Algorithms to Predict Bus Travel Times under Congested Conditions", New Jersey Department of Transportation.
23. Core. (n.d.). by Transitime. Retrieved November 3, 2016, from <http://transitime.github.io/core/>.

24. Daganzo, C. F. 2009. A Headway-Based Approach to Eliminate Bus Bunching: Systematic analysis and comparisons. *Transportation Research Part B: Methodological* 43(10) 913–921.
25. Ding, Y. 2000. “The Development and Application of Dynamic Models for Predicting Transit Arrival Times”, New Jersey Institute of Technology Dissertation, <http://archives.njit.edu/vol101/etd/2000s/2000/njit-etd2000-052/njit-etd2000-052.pdf>.
26. El-Geneidy A., Horning J., Krizek K. 2007. Using Archived Its Data To Improve Transit Performance And Management, Minnesota Department of Transportation.
27. Fattouche Grace. 2007. Improving High Frequency Bus Service Reliability through Better Scheduling, Master Thesis, MIT.
28. Fattouche, Grace. 2011. How to Improve High Frequency Bus Service Reliability Through Scheduling, Proceeding of the ITRN.
29. Federal Transit Administration (FTA). 2010. Transit Operations Decision Support System (TODSS) Core Requirements Prototype Development Case Study and Lessons Learned, US DOT, Washington, D.C.
30. Feng W, Figliozzi M. 2011. Using Archived AVL/APC Bus Data to Identify Spatial-Temporal Causes of Bus Bunching. In *Compendium of Papers of 90th Transportation Research Board*, Washington, D.C.
31. Figliozzi M., W. C. Feng, G. Laferriere, and W. Feng. 2012. A study of Headway Maintenance for Bus Routes: Causes and Effects of Bus Bunching in Extensive and Congested Service Areas. Final Report. Portland State University.
32. Furth P, Muller T. 2006. Service reliability and hidden waiting time: insights from automatic vehicle location data. *Transportation Research Record: Transportation Research Record: Journal of the Transportation Research Board*, Transportation Research Board of the National Academies, Washington, D.C.
33. Furth, Peter G., Brendon Hemily, Theo H. J. Muller, and James G. Strathman. June 2003. TCRP Web Document 23: Uses of Archived AVL-APC Data to Improve Transit Performance Management: Review and Potential. Washington, D.C., Transportation Research Board.
34. Furth, Peter G., Brendon Hemily, Theo H. J. Muller, and James G. Strathman. 2006. TCRP Report 113: Using Archived AVL-APC Data to Improve Transit Performance and Management. Washington, D.C., Transportation Research Board.
35. Gerstle David G. 2012. Understanding Bus Travel Time Variation Using AVL Data. Master Thesis, MIT.

36. Golani H., 2007. Use of Archived Bus Location, Dispatch, and Ridership Data for Transit Analysis, Transportation Research Record: Journal of the Transportation Research Board, Transportation Research Board of the National Academies, Washington, D.C., pp 101-112.
37. Google Developers. Transit. General Transit Feed Specification (GTFS). Accessed July, 2015. <https://developers.google.com/transit/>
38. Google Developers. Static Transit. General Transit Feed Specification (GTFS). Accessed July, 2015. <https://developers.google.com/transit/gtfs/>
39. Google Developers. Realtime Transit. General Transit Feed Specification (GTFS). Accessed July, 2015. <https://developers.google.com/transit/gtfs-realtime/>; <https://developers.google.com/transit/gtfs-realtime/reference>
40. Gurmu, Z. and Fan, W. 2014. “Dynamic Travel Time Prediction Models Using Only GPS Data”, Transportation Research Board Annual Meeting, 14-0378.
41. Hammerle, Meghan, Michael Haynes, and Sue McNeil. 2005. Use of Automatic Vehicle Location and Passenger Count Data to Evaluate Bus Operations: Experience of the Chicago Transit Authority, Illinois. Transportation Research Record: Journal of the Transportation Research Board, No. 1903, Transportation Research Board of the National Academies, Washington, D.C.
42. Heap, R.C., T.H. Thomas. 1976. The Modelling of Platooning Tendencies in Public Transport. Traffic Engineering and Control, Vol. 8 (9), pp.360-362.
43. Heitzman Robert. XLS Tools for Google Transit Feed Specification (GTFS). <https://sites.google.com/site/rheitzman/>. Accessed July 2015.
44. Hounsell N.B., Shrestha B.P., Wong A., 2012. Data Management And Applications In A World-Leading Bus Fleet, Transportation Research Part C: Emerging Technologies, Volume 22, Pages 76–87.
45. Institute of Electrical and Electronics Engineers, Inc. 1998. Software Engineering Standards Committee of the IEEE Computer Society, IEEE Recommended Practice for Software Design Descriptions, IEEE Std 1016-1998.
46. Jeong, R., and Rilett, L.R. 2004. Bus Arrival Time Prediction Using Artificial Neural Network Model, Proceedings of the 2004 IEEE Intelligent Transportation Systems Conference, Washington, D.C., USA, October 3-6.

47. Jeong, R., and Rilett, L. R. 2005. Prediction model of bus arrival time for real-time applications. *Transportation Research Record: Journal of the Transportation Research Board*, 1927(1), 195-204.
48. Kimpel, Thomas J., James G. Strathman, and Steve Callas. 2004. Improving Scheduling Through Performance Monitoring Using AVL and APC Data. Center for Urban Studies College of Urban and Public Affairs Portland State University Portland, Oregon.
49. Kimpel, Thomas J., James G. Strathman, and Steve Callas. 2008. Improving Scheduling Through Performance Monitoring. In *Computer-Aided Systems in Public Transport*, 600:253-280.
50. Kittelson & Associates, Inc., KFH Group, Inc., Parsons Brinckerhoff, Quade & Douglass, Inc., and Katherine Hunter-Zaworski. 2003. TCRP 100: Transit Capacity and Quality of Service Manual, 2nd Edition. Washington, D.C., Transportation Research Board.
51. Kittelson & Associates, Inc., Parsons Brinckerhoff, KFH Group, Inc., Texas A&M Transportation Institute, Arup. 2013. TCRP Report 165: Transit Capacity and Quality of Service Manual, Third Edition. Washington, D.C., Transportation Research Board.
52. Levinson H. 1991. Supervision Strategies for Improved Reliability of Bus Routes (Synthesis of Transit Practice No. 15); Transportation Research Board: Washington, D.C.
53. Lin, J., P. Wang, and D. T. Barnum. 2008. A Quality Control Framework for Bus Schedule Reliability. *Transportation Research Part E: Logistics and Transportation Review* 44.6 (2008): 1086-1098.
54. Mandelzys M, Hellinga B. 2010. Identifying Causes of Performance Issues in Bus Schedule Adherence with Automatic Vehicle Location and Passenger Count Data. *Transportation Research Record: Journal of the Transportation Research Board*, No. 2143, Transportation Research Board of the National Academies, Washington, D.C.
55. MapTools. Tools for Navigation. Latitude and Longitude. Accessed July, 2015. https://www.maptools.com/tutorials/lat_lon/definitions.
56. Mitretek Systems. 2004. Transit Operations Decision Support Systems (TODSS): Core Functional Requirements for Identification of Service Disruptions and Provision of Service Restoration Options 1.0. FTA Office of Mobility Innovation (TRI-10), U.S. DOT ITS Joint Program Office (JPO).
57. Moreira-Matias, L., C. Ferreira, J. Gama, J. Mendes-Moreira, and J. Freire de Sousa. 2012. Bus Bunching Detection by Mining Sequences of Headway Deviation. *Lecture Notes in Computer Science*. Vol. 7377, No. 1, pp. 77-91.
58. Movable Type Scripts. Calculate Distance and Bearing between Two Latitude/Longitude Points Using Haversine Formula in JavaScript. Accessed July, 2015.

<http://www.movable-type.co.uk/scripts/latlong.html>

59. Osuna E. E. and G. F. Newell. 1972. Control Strategies for an Idealized Public Transportation System, *Transportation Science*, Vol. 6, no. 1, pp. 52–72.
60. Pangilinan C., N.H.M. Wilson, A. Moore. 2008. Bus Supervision Deployment Strategies and Use of Real-Time Automatic Vehicle Location for Improved Bus Service Reliability, *Proceedings of 87th Annual Meeting of Transportation Research Board*, Washington D.C.
61. Paul R. Tétreault, Ahmed M. El-Geneidy, “Estimating Bus Run Times For New Limited Stop Service Using Archived AVL and APC Data”, *Transportation Research Board 88th Annual Meeting*, 2009, 19p.
62. Peng, Z., Lynde, E., Chen, W., Kong, C., 2009. Understanding Transit Service Gaps, *Transportation Research Board Annual Meeting*, 2702: p. 19.
63. Reliasoft. 2011. Extreme Value Distributions. *Reliability Hotwire*, the E-Magazine for the Reliability Professionals, Issue 128.
64. Saberi, M, Zockaie K A, Feng, W., El-Geneidy, A. 2013. Definition and Properties of Alternative Bus Service Reliability Measures at the Stop Level, *Journal of Public Transportation*, Volume: 16 Issue Number: 1, pp 97-122.
65. Saberi, M., A. K. Zockaie, W. Feng, and A. El-Geneidy. 2012. Definitions and Properties of Alternative Bus Service Reliability Measures at the Stop Level. *Transportation Research Board 91st Annual Meeting Compendium of Papers*. Washington, DC.
66. Strathman J., 2002. Tri-Met’s Experience with Automatic Passenger Counter and Automatic Vehicle Location Systems, *Center for Urban Studies*, PSU.
67. Strathman, T. Kimpel, S. Callas, and T. Northwest. 2003. Headway Deviation Effects on Bus Passenger Loads: Analysis of Tri-Met’s Archived AVL-APC Data, *Citeseer*, State College, PA.
68. Tétreault P., A. El-Geneidy. 2009. Estimating Bus Run Times For New Limited Stop Service Using Archived AVL and APC Data, *Transportation Research Board 88th Annual Meeting*.
69. TransitWiki. General Transit Feed Specification. Accessed July, 2015. http://www.transitwiki.org/TransitWiki/index.php?title=General_Transit_Feed_Specification
70. Tri-Met. Developer Resources. Tri-Met. Accessed July, 2015. <http://developer.trimet.org/>, <http://developer.trimet.org/GTFS.shtml>

71. Turnquist, M.A. and S.W. Blume. 1980. Evaluating Potential Effectiveness of Headway Control Strategies for Transit Systems, Transportation Research Record, No. 746, D.C.
72. Turnquist, M. 1981. Strategies for Improving Reliability of Bus Transit Service. Journal of Transportation Research Record, 7-13.
73. Turnquist, M. A. 1982. Strategies for Improving Bus Transit Service Reliability. Evanston, IL: Northwestern University. (NTIS Report No. DOT/RSPA/DPB-50-81-27). USDOT, Research and Special Programs Administration.
74. Van Oort N. 2011 Service Reliability and Urban Public Transport Design. Ph.D. Dissertation, Delft University of Technology.
75. Vanchugov, Dimitry. 2012. Intelligent Decision Support Tools that Help Manage, APTA TransiTech Conference, Fort Worth, TX.
76. Welding, 1957. The Instability of a Close-Interval Service, OR, vol. 8, no. 3, pp. 133–142.
77. Wang, L., Zhongyi, Z., and Fu, J. 2014. “Bus Arrival Time Prediction Using RBF Neural Networks Adjusted by Online Data,” Proceedings of The 9th International Conference on Traffic and Transportation Studies, pp 67-75, vol. 138.
78. Wong Alan and Nick Hounsell. 2010. Using the iBus System to provide Improved Public Transport Information and Applications for London. Transportation Research Group, School of Civil Engineering and the Environment, University of Southampton, UK.
79. Yu, B., Lam, W., Tam, M.L. 2011. “Bus arrival time prediction at bus stop with multiple routes”, Transportation Research Part C: Emerging Technologies, Vol. 19, Iss. 6, pp. 1157-1170, December, 2011.
80. Zimmerman, J., Tomasic, A., Garrod, C., Yoo, D., Hiruncharoenvate, C., Aziz, R., Thiruvengadam, N., Huang, Y., and Steinfeld, A. 2011. Field Trial of Tiramisu: Crowdsourcing Bus Arrival Times to Spur Co-design. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI ‘11).

APPENDIX A: General Transit Feed Specification Reference

Term Definition:

1. **Field required** - The field column must be included in the feed, and a value must be provided for each record.
2. **Field optional**- The field column may be omitted from the feed. If an optional column is chosen, each record in the feed must have a value for that column.
3. **Dataset unique** - The field contains a value that maps to a single distinct entity within the column. For example, if a route is assigned the **ID1A**, then no other route may use that ID.

Feed Files

<u>agency.txt</u>	Required	One or more transit agencies that provide the data in this feed.
<u>stops.txt</u>	Required	Individual locations where vehicles pick up or drop off passengers.
<u>routes.txt</u>	Required	Transit routes.
<u>trips.txt</u>	Required	Trips for each route.
<u>stop_times.txt</u>	Required	Times that a vehicle arrives at and departs from individual stops for each trip.
<u>calendar.txt</u>	Required	Dates for service IDs using a weekly schedule.
<u>calendar_dates.txt</u>	Optional	Exceptions for the service IDs defined in the calendar.txt file.
<u>fare_attributes.txt</u>	Optional	Fare information for a transit organization's routes.
<u>fare_rules.txt</u>	Optional	Rules for applying fare information for a transit organization's routes.
<u>shapes.txt</u>	Optional	Rules for drawing lines on a map to represent a transit organization's routes.

<u>frequencies.txt</u>	Optional	Headway (time between trips) for routes with variable frequency of service.
_____	Optional	Rules for making connections at transfer points between routes.
<u>feed_info.txt</u>	Optional	Additional information about the feed itself, including publisher, version, and expiration information.

Figure 2: A Sample of Feed Files

Feed Field Definitions

Following is a list of feed field files, whether they are optional or required, and feed field details.

agency.txt (File Required)

Field Name	Required	Details
agency_id	Optional	The agency_id field is an ID that uniquely identifies a transit agency. The agency_id is dataset unique.
agency_name	Required	The agency_name field contains the full name of the transit agency.
agency_url	Required	The agency_url field contains the URL of the transit agency.
agency_timezone	Required	The agency_timezone field contains the timezone where the transit agency is located.
agency_lang	Optional	The agency_lang field contains a two-letter ISO 639-1 code for the primary language used by this transit agency.
agency_phone	Optional	The agency_phone field contains a single voice telephone number for the specified agency.
agency_fare_url	Optional	The agency_fare_url specifies the URL of a web page that allows a rider to purchase tickets or other fare instruments for that agency online

stops.txt (Required)

Field Name	Required	Details
stop_id	Required	The stop_id field contains an ID that uniquely identifies a stop or station. The stop_id is dataset unique.
stop_code	Optional	The stop_code field contains short text or a number that uniquely identifies the stop for passengers.
stop_name	Required	The stop_name field contains the name of a stop or station.
stop_desc	Optional	The stop_desc field contains a description of a stop.
stop_lat	Required	The stop_lat field contains the latitude of a stop or station.
stop_lon	Required	The stop_lon field contains the longitude of a stop or station.
zone_id	Optional	The zone_id field defines the fare zone for a stop ID.
stop_url	Optional	The stop_url field contains the URL of a web page about a particular stop.
location_type	Optional	The location_type field identifies whether this stop ID represents a stop or station. The location type field can have the following values: 0 or blank – Stop and 1 - Station.
parent_station	Optional	For stops that are physically located inside stations, the parent_station field identifies the station associated with the stop.

wheelchair_boarding	Optional	The wheelchair_boarding field identifies whether wheelchair boardings are possible from the specified stop or station.
---------------------	----------	---

routes.txt (Required)

Field Name	Required	Details
route_id	Required	The route_id field contains an ID that uniquely identifies a route. The route_id is dataset unique.
agency_id	Optional	The agency_id field defines an agency for the specified route. This value is referenced from the agency.txt file.
route_short_name	Required	The route_short_name contains the short name of a route.
route_long_name	Required	The route_long_name contains the full name of a route.
route_desc	Optional	The route_desc field contains a description of a route.
route_type	Required	The route_type field describes the type of transportation used on a route. Valid values for this field are: 0 - Tram, Streetcar, Light rail, 1 - Subway, Metro, 2 – Rail, 3 – Bus, 4 – Ferry, 5 - Cable car, 6 - Gondola, Suspended cable car, 7 - Funicular.
route_url	Optional	The route_url field contains the URL of a web page about that particular route.
route_color	Optional	In systems that have colors assigned to routes, the route_color field defines a color that corresponds to a route. The color must be provided as a six-character hexadecimal number; for example,

		00FFFF. If no color is specified, the default route color is white (FFFFFF).
route_text_color	Optional	The route_text_color field can be used to specify a legible color for text drawn against a background of route_color . The color must be provided as a six-character hexadecimal number; for example, FFD700. If no color is specified, the default text color is black (000000).