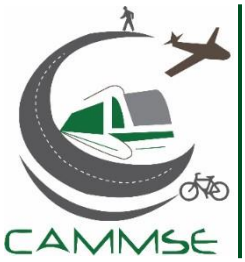


Center for Advanced Multimodal

Mobility Solutions and Education

Project ID: 2020 Project 16

MULTIMODAL CONNECTED VEHICLE PILOT FOR WINTER TRAVEL

Final Report

by

Kakan Dey, Ph.D., P.E. (ORCID ID: https://orcid.org/0000-0002-5875-6180)
Assistant Professor, Department of Civil and Environmental Engineering
West Virginia University, Morgantown, WV 26506
Phone: 304-293-9952; Email: kakan.dey@mail.wvu.edu

Md Tanvir Ashraf, Anthony Carrola
Graduate Research Assistant
West Virginia University

Xianming Shi, Ph.D., P.E. (ORCID ID: https://orcid.org/0000-0003-3576-8952)
Professor, Department of Civil and Environmental Engineering
Washington State University, Pullman, WA 99164, USA.
Phone: 1-509-335-7088; Email: xianming.shi@wsu.edu

for
Center for Advanced Multimodal Mobility Solutions and Education
(CAMMSE @ UNC Charlotte)
The University of North Carolina at Charlotte
9201 University City Blvd,
Charlotte, NC 28223

August 2021

ACKNOWLEDGEMENTS

This project was funded by the Center for Advanced Multimodal Mobility Solutions and Education (CAMMSE @ UNC Charlotte), one of the Tier I University Transportation Centers that were selected in this nationwide competition, by the Office of the Assistant Secretary for Research and Technology (OST-R), U.S. Department of Transportation (US DOT), under the FAST Act. The authors are also very grateful for all of the time and effort spent by DOT and industry professionals to provide project information that was critical for the successful completion of this study.

DISCLAIMER

Table of Contents

List of Figures

List of Tables

EXCUTIVE SUMMARY

Transit service accessibility improvement is an important topic among transit planners and policymakers. Real-time transit information sharing via smartphone app has several benefits, such as lower wait time at the transit stops and higher transit accessibility. Previous studies assessed the benefits of real-time transit information (RTI) sharing mostly by analyzing the survey data or by a simulation-based method. This study collected field trip data to analyze transit trips with a multimodal app (i.e., intervention group) compared to the trips without the app (i.e., control group) to quantify the transit stop wait time reduction in the presence of the multimodal app. The multimodal app disseminated RTI to the users (i.e., location, speed, weather, and road condition information). This study indicates that the average reduction of transit stop wait time after the app use was about three minutes on a selected transit route in Morgantown, WV. The calculated origin-destination-based transit supply index indicates that the network-wide transit supply can be increased by 14% with the use of the multimodal app. The reduced wait time at transit stops reduced the average trip time, and riders can walk from further distances to reach the transit stops. In other words, the long-standing first- and last- mile transit connectivity problem can be improved by using the multimodal app that shares real-time transit information. The findings of this study justify more widescale use of the multimodal app in an urban and rural environment, which can lead to a sustainable transportation system (especially during times of extreme weather).

Chapter 1: Introduction

1.1 Background

Transit service is a sustainable mode of transportation that can reduce congestion, alleviate negative impacts of automobiles on the environment, and provide equitable transportation services (*1*). The determination of transit service accessibility is an important issue for proper planning and improving the transit ridership. There are several ways to measure transit accessibility: stop count, coverage-based, frequency-based, and gravity-based measures (*2*). Among these, coverage-based measures, also known as transit service catchment area, are the most widely used to measure transit accessibility. A transit stop catchment/service area is defined as the area around a transit stop where transit users can access that stop comfortably (*3*). Areas beyond the catchment area of a transit are considered to have no effect on the ridership of that stop. Quantification of the transit stop coverage area largely depends on the spatial and temporal characteristics of the geographic area where the stop is situated.

The spatial boundaries of transit coverage areas are related to the long-standing first- and last-mile connectivity problem (*4*). During the first and last mile of a trip, people usually walk or use bicycles to get to the transit stop (*5*). Many previous studies explored the walking distance to the transit stop (*6-10*). Because of the uncertainties arising from traffic congestion and inclement weather conditions, transit route delays and route changes or cancellations are common phenomena undermining the accessibility and mobility at the transit stops. Bus stop waiting time can be significantly reduced if the users are more informed about these uncertainties, which can intuitively increase the available network capacity as well as transit stop coverage area (*10*).

Connected Vehicle (CV) applications such as a multimodal app that disseminates real-time transit information to the users/riders have the potential to increase transit accessibility. CV technologies improve traffic flow and reduce congestion and time delays in the roadway system, which is highly desirable for effective and efficient transportation systems (*11*,*12*). CV applications can also impact transit services by increasing ridership, decreasing stop wait time, and decreasing the overall travel time of the passengers. Quantifying these transit services and their network-wide temporal distribution in the presence of CV application is critical for the informed transit planning and operations decisions. These analyses can also help the planners

justify the mass deployment of CV in transit services. However, none of the previous studies assessed the benefits of multimodal CV application in transit services in terms of accessibility improvement and network-level transit supply improvement.

1.2 Research Questions

The research questions that were investigated in this study are: (i) Does the multimodal app increase transit accessibility? (ii) To what extent the transit service areas can be expanded if the multimodal APP is introduced to the users to facilitate the non-motorized accessibility to the transit? (iii) Does the APP effectively reduce the transit stop wait time?

1.3 Expected Contributions

This study developed a smartphone-based multimodal app to collect real-time transit information, weather and roadway condition data, and user's trip information to answer these research questions. The relevant data, which includes real-time bus location, average bus speed during each trip, estimated arrival time of the bus, and distance to the transit stop for the first connectivity, was then displayed in the app to the transit users. Improvement in transit accessibility was measured by quantifying the expansion in transit coverage area compared to the baseline scenario (i.e., without app scenario). In addition, the increase in the available capacity of the transit routes based on the reduction of transit stop waiting time was measured.

1.4 Report Overview

The rest of this paper is organized as follows: the literature review section provides a comprehensive overview of the past studies focused on transit accessibility measures, followed by the gap in the existing literature and the contribution of this study. The following three sections explain the adopted methods for quantifying transit accessibility improvements, study design for field data collection, the results of the accessibility improvements, and potential implications of the findings. The last section presents the concluding remarks.

Chapter 2: Literature Review

Transit accessibility improvement has been an important research topic among transit planners and policymakers in the last few decades. Generally, accessibility is measured as the ease of accessing locations/services (*13,14*). Transit accessibility is classified into three main categories: (i) access to transit stops or system accessibility, (ii) system-facilitated accessibility, and (iii) access to destinations (*15*). The first two categories describe network access or access provided by a transit facility to travel to a destination. The third category represents the number of possible destinations that can be accessed from an origin using public transit (*16,17*). System accessibility models usually deal with physical access to public transit stops by evaluating the distance, time, and effort to reach a transit stop (*15*). Thus, system accessibility measures are related to the first-mile connectivity of transit stops.

The first-mile access to transit services can be measured using routes or stops as the reference of measurement (*4*). However, stops are more appropriate references than routes as people use bus stops to get into or out of the transit modes (*18*). Studies in the past used several distances to define the catchment area of public transit stops (*19-21*). A common value for defining transit stop coverage area is 0.25 miles (or 400 m), equivalent to five minutes of walking distance of a pedestrian (*8,9*). Guerra et al. (*22*) suggested the use of a 0.25-mile pedestrian-transit catchment area around transit for work-related trips and a 0.5-mile catchment area for residence. In another study, El-Geneidy et al. (*6*) found that the 85th percentile of walking distance to transit stops was about 0.5 km for home-based bus trip origins and 1.3 km for home-based commuter rail trip origins. Ma et al. (*21*) and Ashraf et al. (*23*) used a 0.25 miles catchment area around Metrorail Station in Washington DC, and Subway stops in New York City to define the catchment area. Similarly, Noland et al. (*20*) used 0.25 miles buffer area around bike-sharing stops to calculate their service areas.

Previous studies explored the transit stop catchment area based on the data collected from traditional travel surveys and Origin-Destination surveys (*6,24*). Studies based on the travel surveys cannot identify transit stop catchment areas with sufficient accuracy, as they are self-reported and do not track/collect actual walking distance data (*4*). Some previous studies used mobile phone trip data and Global Positioning System (GPS) data containing detailed trip

trajectories to identify the transit stop catchment area (*25,26*). The GPS data was used to determine the locations users visited and their origins. This actual trajectory data can be easily used to identify the true coverage area of a transit stop. Some studies used the as-the-crow-flies distance to delineate the geographical boundary of the transit stop coverage area (*8,9,27*). In addition, some studies used distance based on the actual street network to identify the coverage area (*6,18,28*). While the transit stop coverage area calculation based on the as-the-crow-flies distance is easy to estimate, it is problematic and overestimates the coverage area, and network-level distance is more appropriate than the as-the-crow-flies distance.

The transit stop coverage area can increased by introducing CV applications that disseminate real-time transit information to the users. During inclement weather in the winter season, roadway capacity gets reduced, increasing overall travel time for the impacted routes (*29*). CV applications such as multimodal app can effectively disseminate transit vehicles' real-time locations, roadway weather conditions, and transit delay information to the users, allowing them to make an informed decision about their trips. With this application, the transit stop wait time can be reduced significantly, enabling the users to travel extra distance at the beginning or end of their trips.

Real-time information (RTI) sharing with users about transit vehicle location, speed, and arrival time has become common among transit operators. Studies in the past found that RTI sharing with passengers has several benefits, including a decrease in transit stop wait time, reduction in overall trip time, and increased transit ridership (*30-32*). However, most of the studies related to decrease in transit stop wait time were either based on the survey asking respondents about the stop wait times (*10,33,34*) or based on a simulation model (35,36). Brakewood et al. (*37*) evaluated the impacts of RTI sharing on the bus riders in Tampa, Florida. They performed a behavioral study among the bus riders by dividing them into control group and treatment group where only treatment group used the RTI. They used a web-based survey to measure the behavioral, feeling, and satisfaction survey of the treatment group versus the control group. They found a significant decrease in wait time at the bus stops among the treatment group users compared to the control group. A 16% reduction in wait times among the RTI user groups was reported.

Some of the previous studies also explored the benefits of RTI in terms of ridership increase of transit modes. Two consecutive studies were done by Tang & Thakuriah in the year 2011 and 2012 in Chicago, Illinois on the benefits of RTI (*38*,*39*). The first study used a stated preference survey and found that 76.1% of respondents would increase transit use if RTI is available (*38*). The second study used the econometric modeling technique to estimate the bus trips and found an increase of 126 average weekday bus trips per route due to RTI (*39*). Based on the findings of past studies, it is clear that RTI sharing with the users can increase transit accessibility by decreasing wait time at transit stops and increasing user's willingness to use transit services. With the increase in users' willingness to use the transit services more frequently, there might be an increase in the number of transit users who will walk or bike a longer distance to access the transit services and increase the transit stop's first and last-mile coverage area. However, none of the previous studies explored the transit stop coverage improvement in the presence of RTI sharing with the user via the multimodal app, and real-time GPS tracking data to determine walk time, stop waiting time, and overall trip time.

The present study deployed a multimodal app to disseminate RTI and roadway weather information to the users and collected detailed GPS trajectory data of the trips. Based on the GPS data, O-D of the trips and transit stop wait time were determined. This wait time of the treatment group was compared with the wait time of the control group. Based on the literature review findings, RTI and road weather information sharing should decrease the transit stop wait time. Due to the reduction in transit stop wait time, the stop-wise coverage area increase was determined. Furthermore, the reduction in wait time at transit stops should increase the average travel speed for each trip which subsequently increases the available capacity of the routes. The improvement in available route capacity was also quantified in this study.

Chapter 3: Method

3.1 Introduction

This section presents the study design of the field tests along with the analysis methods and formulation of the Origin-Destination based supply index. A brief description of the APP design is presented in subsection 3.2. Definitions of the subway stop catchment area, stop buffer area ratio, and total available capacity is provided in section 3.3, 3.4, and 3.5, respectively. Finally, the experimental setup was included in section 3.6.

3.2 Multimodal App Design

The multimodal app was designed to disseminate RTI (i.e., location of the bus, estimated arrival time, delay, the available and total capacity of the bus, and bike racks) to transit users. In addition, the multimodal app provides real-time weather information and road surface condition information to users. Figure 3.1 presents the high-level architecture of the multimodal app.

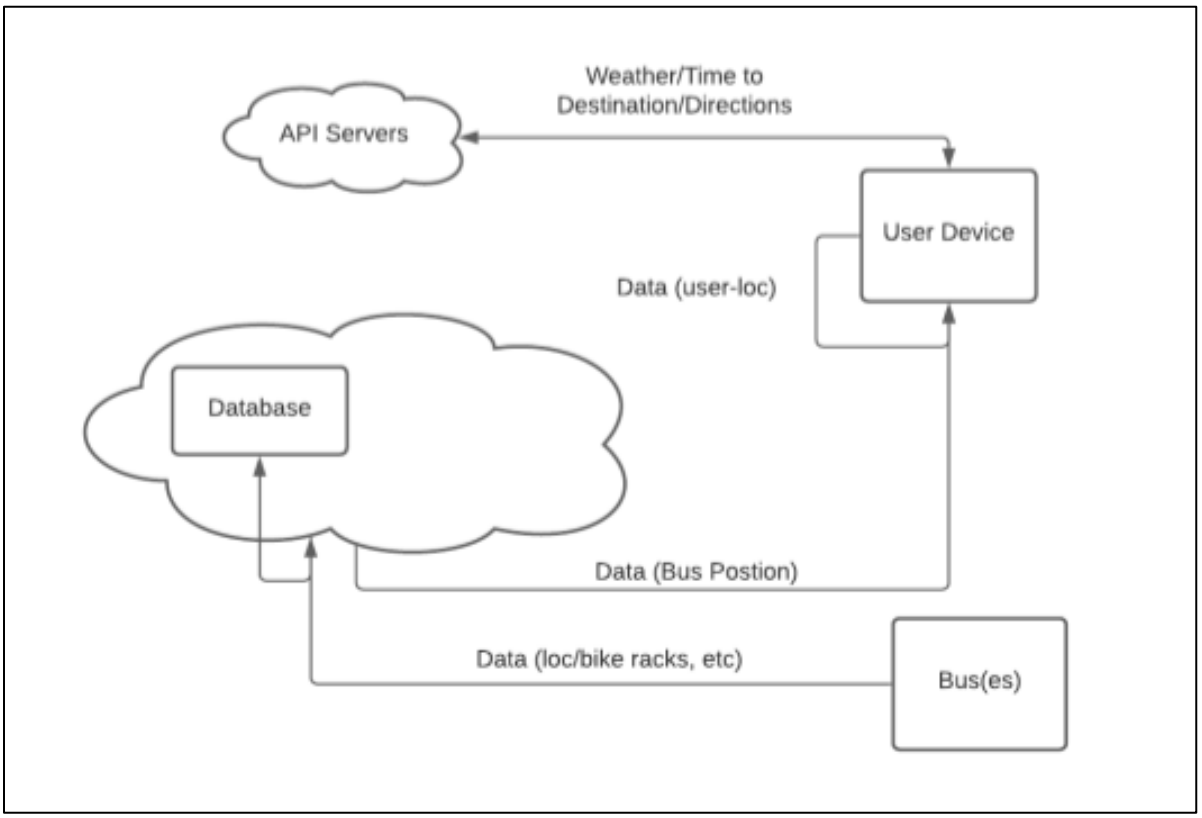


Figure 3.1: High-level architecture of the multimodal app

The dynamic bus information such as location and capacity were collected by a separate app called bus app that was present in the transit bus. As shown in the app architecture, the bus app sent data (i.e., real-time location, speed, and capacity) to the server where this dynamic information was stored. The multimodal app collected the most recent bus information from the server and displayed it to the user. The multimodal APP also collected user location and speed data via the user device (e.g., an android smartphone or other android device with a cellular connection and a GPS), which were then used to calculate the estimated walk time to reach a bus stop. The weather information was collected using the Open Weather API and updated every two minutes. In addition to disseminating real-time transit and weather information, the multimodal app also tracked individual trips from each user and randomly assigned a trip number that was used later to extract detailed trip data (i.e., speed, location, and time) from the server. In this study, detailed trip data collected by the multimodal app was used to perform the analysis.

3.3 Catchment area

A transit stop coverage/catchment area was measured as the zone within the study area from where transit service can be accessed by bicycle or walking (*4*). In this study, only walk mode was used for the first- and last-mile connectivity. A distance of 400 meters or a quarter-mile radius from a transit stop, which is equivalent to a five-minute walking distance of a pedestrian, was used to define the catchment area. The transit stop catchment area was calculated using the existing roadway network facilities in Morgantown, WV area.

For the baseline scenario (i.e., without multimodal app trips/ control group), the catchment area was calculated by utilizing the Network Analysis tool in ArcGIS pro software using travel time as the cost function. For the treatment group (i.e., with multimodal app trips), the connectivity time was increased, and expansion of the service area was quantified.

3.4 Stop Buffer Area Ratio (SAR)

Transit stops are the locations where people board in or drop off from the transit buses. The spatial distribution of the transit stops along the routes depends on the demographic characteristics of the zones. Larger and more densely populated areas need more stops compared to less densely populated and smaller zones (*40*). Transit supply can be defined based on the portion of the area or population served by the transit services. In this study, Stop Buffer Area

Ratio (SAR) defined by Hussain et al. (*40*), was used to determine the increase in transit catchment area due to using the multimodal app. The SAR can be expressed as follows:

$$SAR_i = \frac{\sum_{n=1}^{N} Area_{B_{n,i}}}{Area_i} \quad (3.1)$$

Where $\sum_{n=1}^{N} Area_{B_{n,i}}$ is the summation of the portion of the buffer areas from the stop 'n' falling into the zone 'i'. In the equation, the overlapping buffer area of two stops within the zone is counted twice as the overlapped area of the zones is served by two separate stops and offers enhanced connectivity (*40*). Figure 2 illustrates the calculation of the stops buffer area ratio. For example, in figure 2, two transit stops (Evandale Crossing and Towers) buffer areas intersected census tract 120. The red area marked in Figure 3.2 represents the portion of the census tract, which is overlapped by two separate transit stops buffer areas. This area represents an enhanced opportunity for transit access. Polygon abcde shows the portion of the buffer area of the bus stop located at Towers that intersected with census tract 120. Similarly, polygon efghijd represents the portion of the bus stop buffer area located at Evansdale crossing that intersected with census tract 120. The SAR was calculated using transit stop buffer areas that were found before and after the introduction of the multimodal APP. As transit users can walk from longer distances to get to the bus stops after the multimodal app was introduced, the transit stop buffer zone should be larger than without a multimodal app scenario and result in higher SAR values. An increase in SAR values indicates an increase in transit supply for the zones.

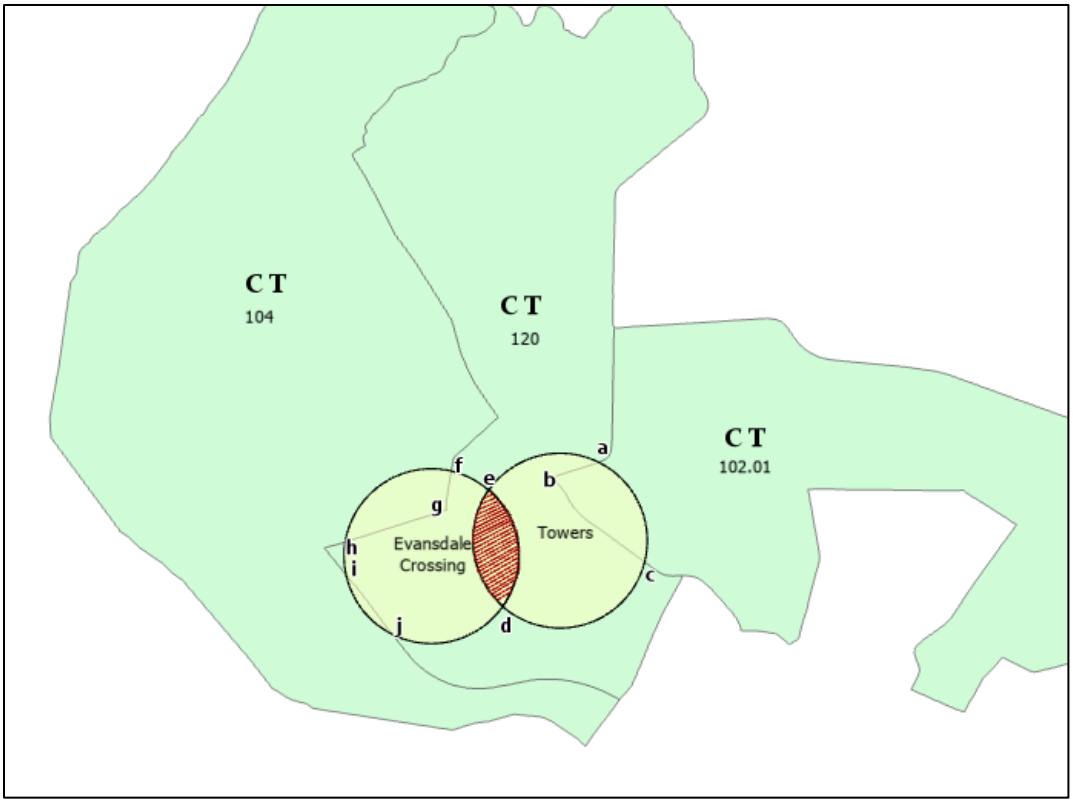


Figure 3.2: Illustration of Stops Buffer Area Ratio (SAR) Calculation

3.5 Total Available Capacity of the route

To identify the total available capacity of the transit routes, an Origin-Destination based Supply Index (odSI) developed by Hussain et al. (*40*) was adopted in this study. Hussain et al. (*40*) defined the odSI as the spatial and temporal availability of transit services and its quality while traveling from one origin zone to another destination zone. The odSI can be presented as follows:

$$odSI_{ij} = \sum_{r=1}^{R}\left[S_r \times \underbrace{\frac{1}{e^{N_{t,r}}} \times \min\left(\frac{\bar{\vartheta}_{ij,r}}{V_f}, 1\right)}_{\text{Route Connectivity}} \times \underbrace{\sum_{n=1}^{N_r}\left(\frac{Area_{B_{n,i}}}{Area_i} \times \sum_{t=1}^{T}\max\{(TC_r - O_{n,t_r}), 0\}\right)}_{\text{Service Availability}}\right] \quad (3.2)$$

Where S_r is the average straightness ratio of the route, r; $N_{t,r}$ is the minimum number of transfers required to travel from origin zone i to destination zone j; $\bar{v}_{ij,r}$ is the average transit speed between origin i to destination j; V_f is the desired transit speed; TC_r is the total capacity of

the transit route; and $O_{n,t}$ is the occupancy of the route r at stop n. Once the multimodal app is introduced, the SAR value and the average trip speed between the origin and destination zones are expected to increase. From equation 2, the odSI is directly proportional to the average transit speed and SAR values which means that with an increase in average trip speed and SAR values, the odSI values should also increase. An increase in the odSI value means that the available capacity of the route increased due to the introduction of the multimodal app. The odSI value will be used to estimate the benefits of using the multimodal app and measure the increase of available transit route capacity.

3.6 Experimental Setup

Field data collection trips were conducted to estimate the effectiveness of the multimodal app in terms of travel time reduction, transit service area increases, and an available capacity increase of the transit services. The study design and implementation of the field tests are presented below:

The primary objective of the field tests was to determine the benefits of using the multimodal CV application. Ten participants were recruited to participate in this study. The app was designed to automatically track the users' movements, which can be used to extract trip information such as trip start and end location, speed, wait time at the stops. As people aged between 21-30 years uses the smartphone more frequently for their daily purposes (*41*), all recruited participants were between 18-35 years of age. Total 10 young adults living in the West Virginia University Morgantown campus area participated in the field data collection. Previous literature showed that, a small number of participants is sufficient for an exploratory study where the goal of the study is to investigate and explain phenomena as consciously experienced, which justifies the small number of participants used in this study (*33*,*42*). Among the selected participants, one group of participants completed their daily trips along the chosen route without using the multimodal app. In this case, they only knew about the transit stop-wise scheduled arrival time of the buses. This group was called the control group. The other group of participants with the multimodal app were asked to plan and complete their trips using the app. This group of participants was called the intervention group. The trip trajectory data, average trip speed, wait time at the transit stops, and first- and last-mile connectivity time were collected for each trip. Trips were performed at different times of the day (peak hour and off-peak hour) and

different weekdays to capture the temporal effect. The sample size was determined based on the average travel time standard deviation of 4 minutes. For a tolerance of 1 minute and a 90% confidence interval, the estimated sample size was 44. The research team collected data about 100 trips data among which 50 trips were from the control group, and 50 trips were from the intervention group. Previously, Dziekan and Vermeulen (*43*) and Brakewood et al. (*37*) used a similar sample size to analyze RTI on passenger wait time. Also, participants were asked to use the walk as their mode for the first- and last-mile connectivity as using bicycles was not feasible for the experimental setup.

Chapter 4: Field Data Collection

4.1 Study Area and Route Selection

West Virginia University (WVU), situated in Morgantown, West Virginia, was selected as the study area. This study was conducted using tablets on buses and multimodal smartphone app for users to collect GPS location, travel speed, and other trip information using a cellular communication network. In Morgantown, Mountain Line Transit Authority (MLTA) is the only transit service provider. Among the routes operated by MLTA, Route 38 is one of the most frequently used routes that run in between the Engineering Campus and the Downtown Campus of WVU. In this study, Route 38 was used to implement the multimodal app for the field data collection. The study area and the selected route are shown in Figure 4.1.

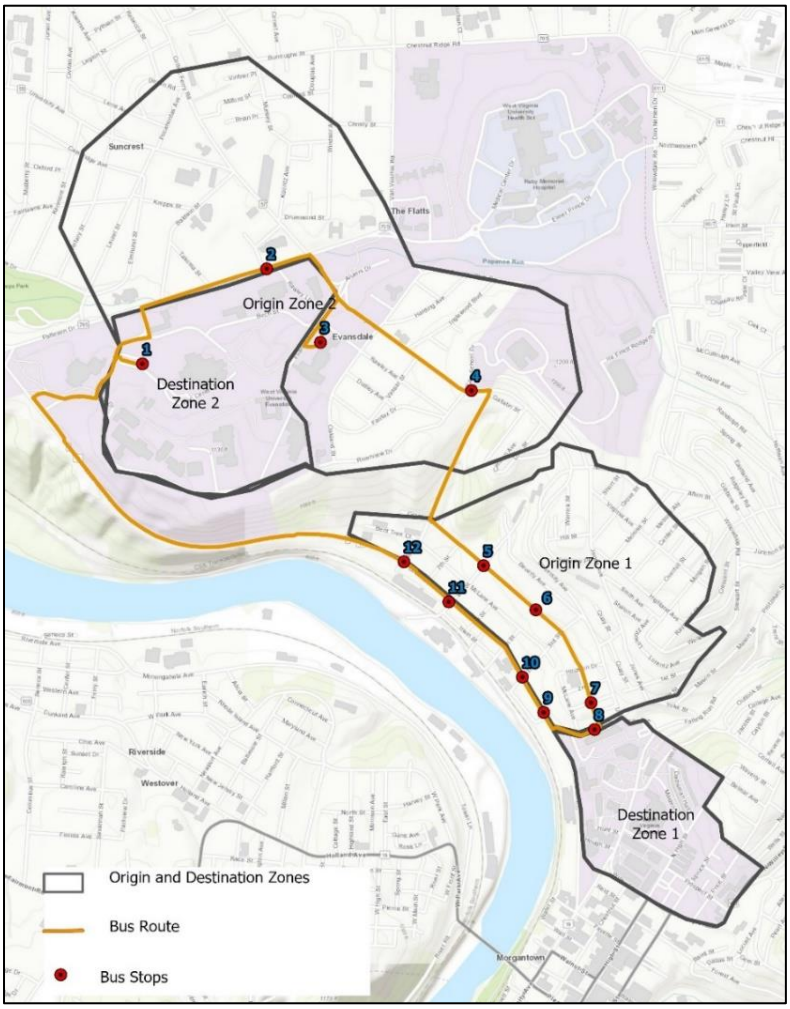


Figure 4.1: Study area and Selected route for case study application

4.2 Route Characteristics

Route # 38 known as “Blue and Gold” runs between the Evansdale Campus and the Downtown Campus during the weekdays (Figure 3). The WVU students primarily use this route to travel between these two campuses. The total length of the route is about 4 miles with a headway of 20 minutes. There were twelve (12) stops along this route which is shown in Figure 3.

4.2.1 Average first-mile walk distance

In this study, walking was used as the only non-motorized mode to access (first mile) the transit stops. As shown in Figure 4, the walking distance to access the transit stops at the origin zones ranges from 0.1 miles to 0.6 miles based on the collected trip data. The average walking distance for these trips in the origin zones is 0.275 miles. It is evident from Figure 4.2 that, with the increase in walking distance, the number of trips decreases.

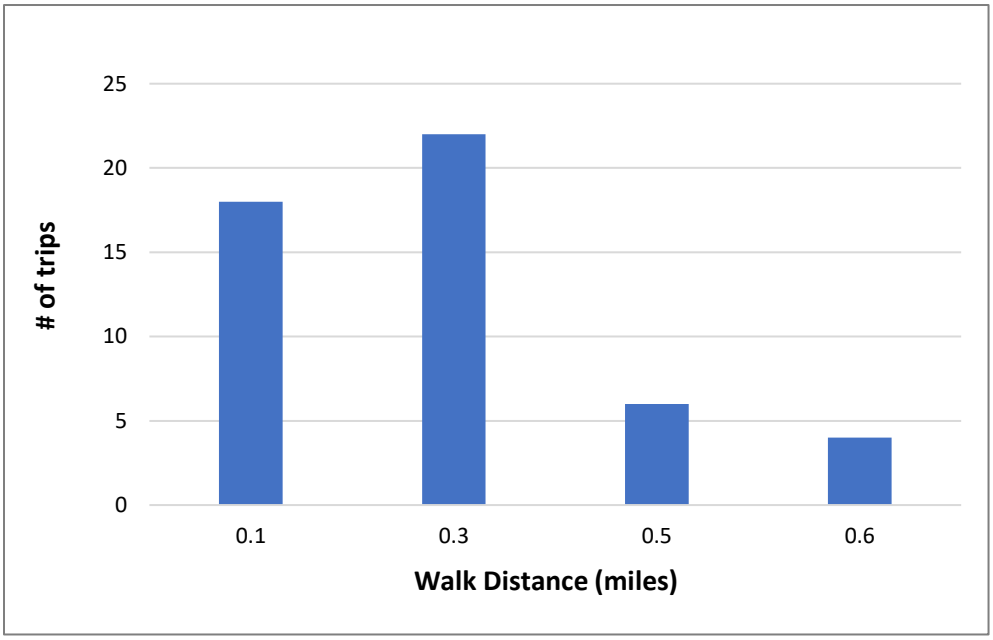


Figure 4.2: First-mile walking distance for the trips using the multimodal app

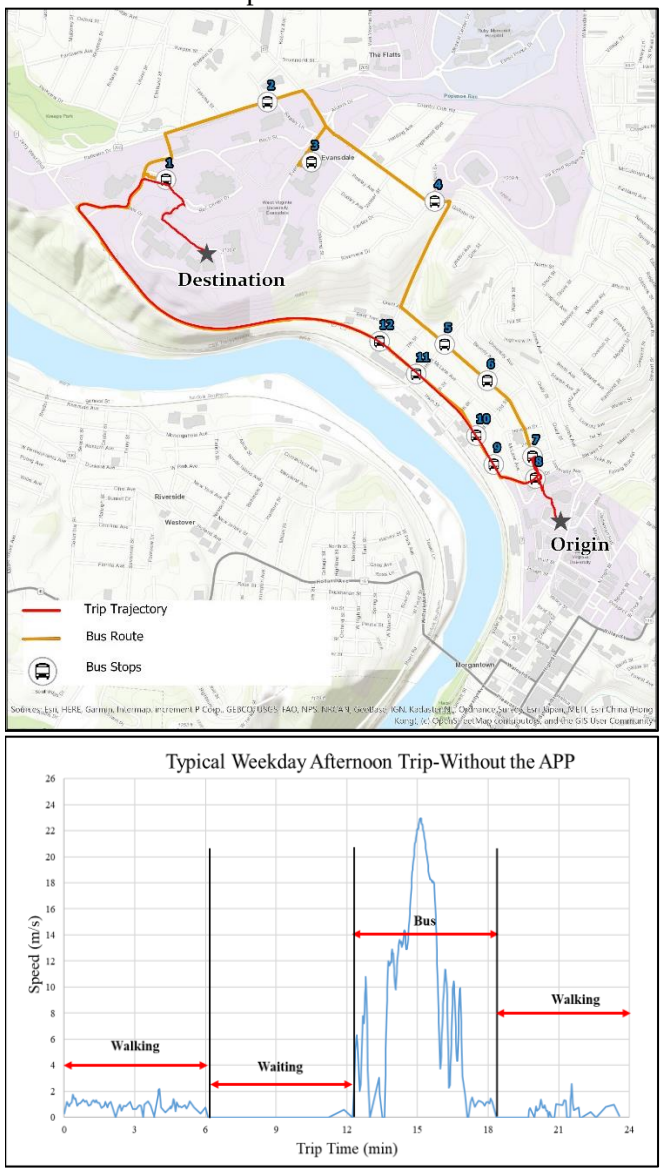
4.2.2 Speed Distribution

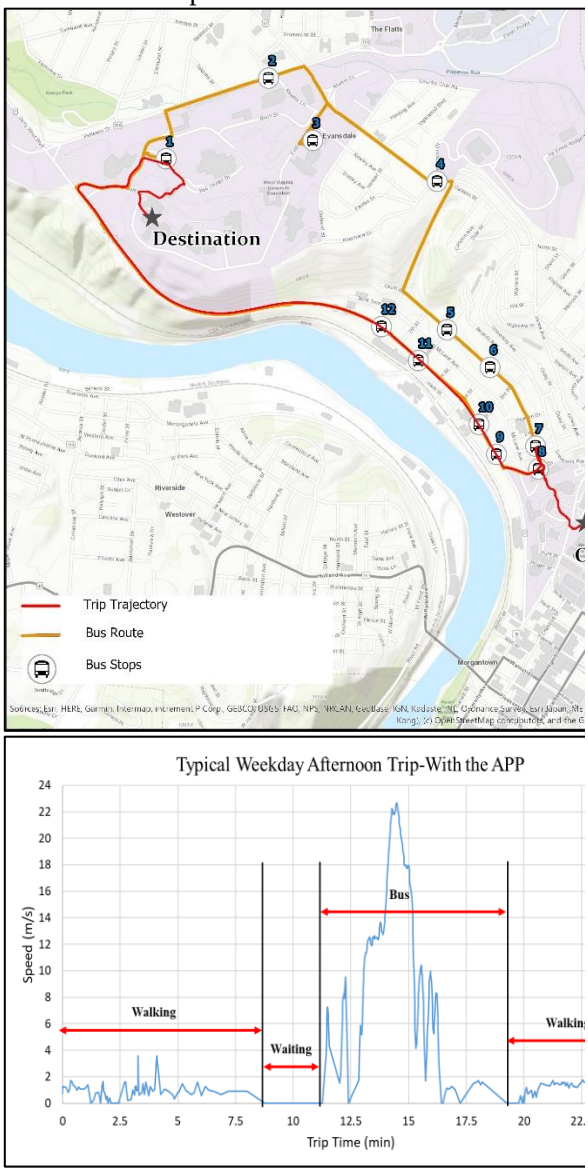
For the study purpose, average trip speed was automatically collected using the multimodal app. Figure 5 represents the trip trajectory and speed distribution of two typical weekday afternoon trips with walking as the first-mile connectivity mode. Figure 4.3a shows the

speed distribution of the trip that did not use the multimodal app, and Figure 4.3b shows the speed distribution of a trip that used the multimodal app. Transit stop wait time for each trip was calculated using the speed profile. The speed profile in Figure 4.3a shows that, after getting to the bus stop by walking, the speed was zero for about 6 minutes which is the bus stop wait time. After that, there was a speed increase with a peak value of 51.5 mph. This portion of the trip was traveling in the transit bus, followed by walking again for the last mile connectivity to the destination. For another trip (shown in Figure 4.3b) with the same origin where the user used the multimodal app for the RTI, the wait time was significantly lower than the trip without the multimodal app. For this trip, the first-mile connectivity time for walking was 8 minutes, and the transit stop wait time was 3 minutes compared to 6 minutes waiting time for without app trip.

Trip A: Walk-Bus-Walk
Start time: 2:11 PM
End time: 2:35 PM
Stop wait time: 6 mins
Trip duration: 24 mins

Trip B: Walk-Bus-Walk
Start time: 11: 52 AM
End time: 12: 17 PM
Stop wait time: 3 mins
Trip duration: 25 mins





(a) **(b)**

Figure 4.3: Speed Distribution of (a) Typical Weekday Afternoon trip without app; (b) Typical Weekday Afternoon Trip with app

Chapter 5: Case Study Result and Discussion

5.1 Introduction

The benefit of the multimodal app was assessed based on three performance measures of the route and its associated stops (discussed in the methods section), which are Stops Buffer Area Ratio (SAR), Service Area of the stops, and O-D Supply Index (*odSI*). The following section discusses the results from the control group vs. intervention group trips.

5.2 Stops Buffer Area Ratio (SAR)

Transit supply depends on the proportion of an area that is served by the transit services (*40*). An increase in the proportion of area served by transit stops indicates an increase in transit supply. At first, a reduction in transit stop wait time was calculated to quantify the improvement in transit supply after the multimodal app was introduced. The transit stop wait time reduction was calculated using the trip data from the control group and the intervention group. For the control group, the average wait time at transit stops was 4 minutes 37 seconds with a standard deviation of 1 minute 57 sec, where the average transit stop wait time was 1 minute 43 seconds for the intervention group with a standard deviation of 1 minute. The statistic suggests that there could be a 2 minutes 55 seconds reduction in average transit stop wait time which was significant with p-value close to zero. Previous studies found that wait time had a negative impact on first mile access distance to transit stops (*44*). El-Geneidy et al. (*44*) reported that people tend to walk longer distance when the transit services had shorter wait time. As such, the reduction in wait time can allow the transit users to reach transit stops from a further distance. Table 5.1 represents the calculated Stops Buffer Area Ratio (SAR) for origin zone 1 and 2 (Figure 4.1). From Table 5.1, it is evident that once the multimodal app was introduced, the SAR value increased for both origin zone 1 and 2. For origin zone 1, the SAR value was 2.78 for the control group trips and 3.03 for the intervention group trips, equivalent to a 9% increase in stops buffer area ratio. Similarly, the SAR value for the control group trips was 1.09 and for the intervention group trips the SAR value was 1.34 for origin zone 2, which is equivalent to a 23% increase in the stop buffer area ratio.

As transit services are accessed at transit stops, the service area of each stop needs to be expanded to increase transit accessibility. As seen from the field trip data, the transit stop buffer area got larger when the multimodal app was used indicating an increase in transit stop

accessibility. Transit accessibility can be increased by establishing new transit infrastructures (e.g., new routes/stops); they are usually resource-intensive (e.g., construction and maintenance cost). However, the multimodal app can be used to increase transit accessibility and increase the network-wide transit supply in a low cost (e.g., app development and maintenance cost) and sustainable manner.

Table 5.1: Comparison of the Stop's buffer area ratio for the selected route

Zone	Origin Point	$Area_i$	$Area_{Bn,i}$	Stops' Buffer Area Ratio (Before)	Stops' Buffer Area Ratio (After)
1	Stop 5	0.237	0.114	2.78	3.03
	Stop 6		0.133		
	Stop 7		0.0855		
	Stop 8		0.076		
	Stop 9		0.076		
	Stop 10		0.055		
	Stop 11		0.065		
	Stop 12		0.055		
2	Stop 1	0.61	0.13	1.09	1.34
	Stop 2		0.19		
	Stop 3		0.19		
	Stop 4		0.16		

5.3 Coverage Area Improvement for the Route

The transit stop buffer area ratio was calculated by creating a circular buffer area around each transit stop. Although this method is quick and easy to calculate, it overestimates the stop level buffer area (*4*). The total coverage for the route was calculated using the network distance method using the existing road network of Morgantown, WV to avoid this issue (*4*). During the process, travel time was used as the cost function. As discussed in the previous section, the field trip data showed that the transit stop wait time was reduced by around three minutes using the multimodal app. So, the service areas before and after introducing the multimodal app were calculated using five minutes and eight minutes as the first-mile travel time and using walking as the travel mode. The resulting service area map is shown in Figure 5.1.

For the control group trips without the multimodal app, the total length of the road network covered was 32 miles, denoted by the red lines in Figure 5.1. On the other hand, for the intervention group trips with the multimodal app for acquiring RTI, the total length of the road

network increased to 44 miles, equivalent to a 37% increase compared to without the multimodal app scenario (extra road network is denoted by blue lines).

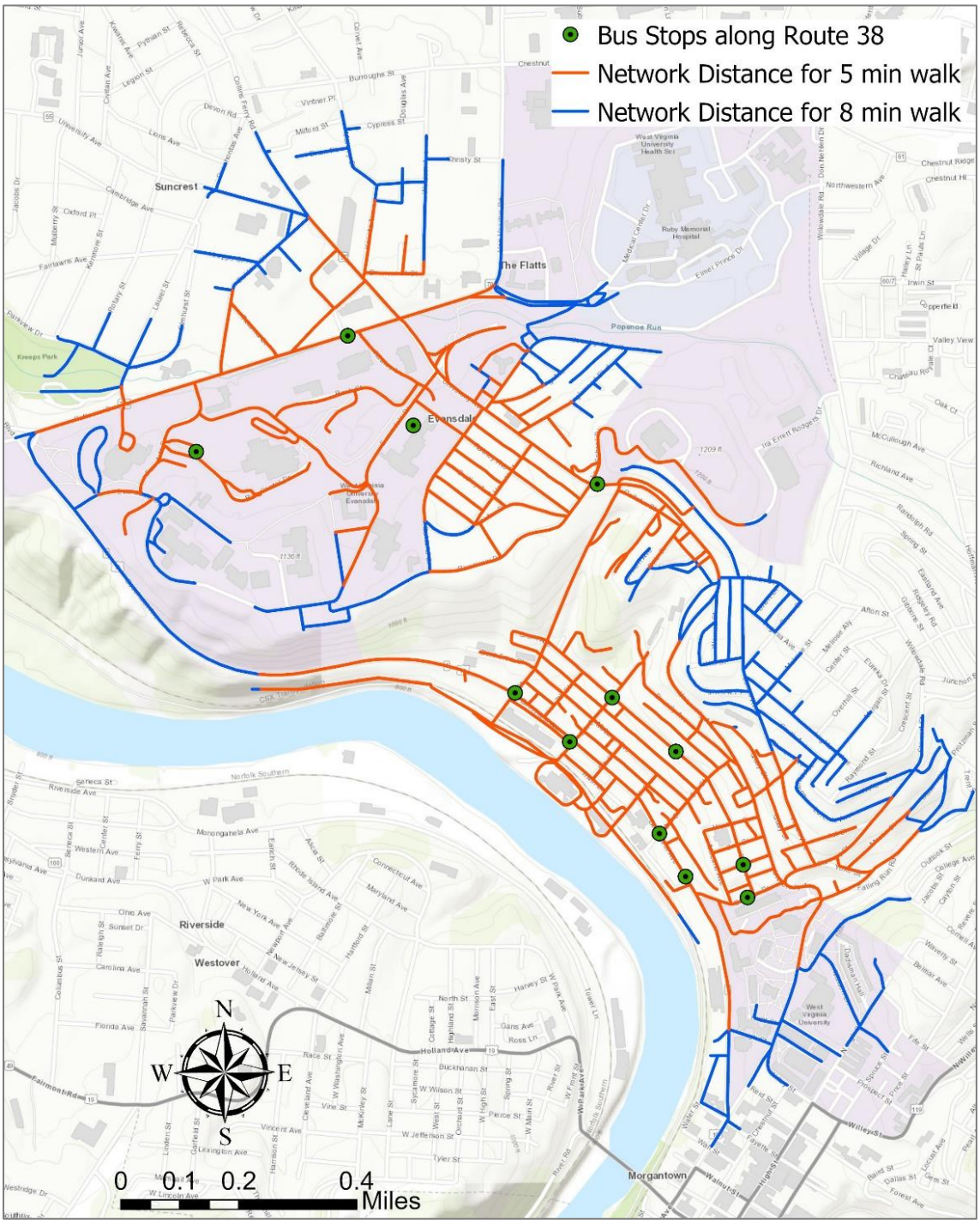


Figure 5.1: Service area improvement after the use of the multimodal app

5.4 Transit Supply Index (odSI)

The *odSI* index was calculated for the selected route using both control group trips and intervention group trips based on the formula (Equation 2), presented in the methods section. The index is based on the Origin and Destination (O-D) pairs. The Origin zones are defined based on the student residential areas along the route and two campuses (i.e., Evansdale campus and Downtown campus). For the field data collection purpose, two Origin zones (Zone 1&2) were defined, as shown in Figure 4.1. The Destination zones were defined as the two Campuses (Evansdale and Downtown) of WVU. The route connects the Origin and Destination zones directly (i.e., no transfers were required for the trips). The route straightness ratio in equation 2 is defined as the ratio of shortest distance (using car or bus) to the optimal transit route distance between certain O-D pairs (*40*).

Routes with a higher straightness ratio indicate direct access to destinations and are relatively more popular than routes with a lower straightness ratio. For the O-D pairs in this study, the shortest distance was equal or close to the optimal transit route distance. The route straightness ratio was used as 1 to calculate the *odSI*. Eight stops serve origin zone 1 (stops 5-12), and Origin zone 2 is served by four stops (stops 1-4), as shown in Figure 4.1. The total buffer area for a O-D pair is the sum of the individual transit stop's buffer area that falls within the Origin zone of that O-D pair. In intervention group trips, the average speed increased as the stop wait time was less than the control group trips. Finally, to calculate the available capacity at each transit stop, it was assumed that the bus was 50% occupied of its full capacity. Assuming the full capacity of 40 passengers (i.e., a typical transit bus), the total available capacity of the route for Origin zone 1 to destination zone 2 was 160 as eight stops were located in origin zone 1 (Figure 4.1). In comparison, for trips starting from origin zone 2 ending in destination zone 1 the total available capacity was 80 as the number of stops that falls within in origin zone 2 is four (Figure 4.1).

Table 5.2: Summary of the calculated parameters and results of the odSI calculation

Scenario	O-D pair	Average Speed (mi/hr)	Minimum Transfers	Straightness Ratio	Total Buffer Area (Sq mile)	Zone Area (Sq mile)	Total Available Capacity	SI provided by the route
Before	1-2	16.29	0	1	0.6595	0.237	160	362.64
	2-1	11.96	0	1	0.67	0.61	160	105.09
After	1-2	18.45	0	1	0.6595	0.237	80	410.73

	2-1	13.71	0	1	0.67	0.61	80	120.47

Using the values of the parameters (shown in Table 5.2), the estimated value of the *odSI* without the use of the multimodal app (control group) was 362.64 and 105.09 for O-D pair 1-2 and 2-1, respectively. On the other hand, with the use of the multimodal app, the *odSI* value increased to 410.73 and 120.47 for O-D pair 1-2 and 2-1, respectively. The increase in *odSI* value indicates that with the multimodal app's introduction, the transit supply increased for both the O-D pairs. With increased supply, the transit network will be able to meet future increase in transit ridership demand, making the transit system more sustainable. The *odSI* value is directly proportional to the average speed value of the trips and the stops buffer area ratio. These parameters (average speed and stops buffer area) increased for the intervention group trips, contributing to the increase in *odSI* values.

5.5 Multimodal App- System-wide Impact Estimation

System-level impact of the multimodal app was estimated based on the stop wait time savings from the test route. Mountain Line Transit Authority (MLTA) currently has 24 routes that operate around the Morgantown area in West Virginia (*45*). One route that connects to Pittsburgh, Pennsylvania, was excluded from the analysis to estimate the systemwide impacts in the Morgantown area only. The stop wait time savings due to the multimodal app were applied to all routes to calculate systemwide stops buffer area ratio and service area calculation. Figure 5.2 represents the calculated service area of the bus stops for the 23 MLTA routes. Census Tracts were used as the origin zones to calculate stops buffer area ratio calculation. The brown polygon in Figure 5.2 depicts the 5 minutes walking area, and the green polygon depicts the additional area that can be reached in eight minutes due to a reduction in stop wait time. The systemwide result reveal that introduction of the multimodal app could increase MLTA coverage area and nearly double population coverage compared to without app scenario. The average SAR value before the multimodal app use for all census tracts was 0.34, and after the multimodal app use was 3.34. The increase in SAR also indicates increased overall transit accessibility for all the routes.

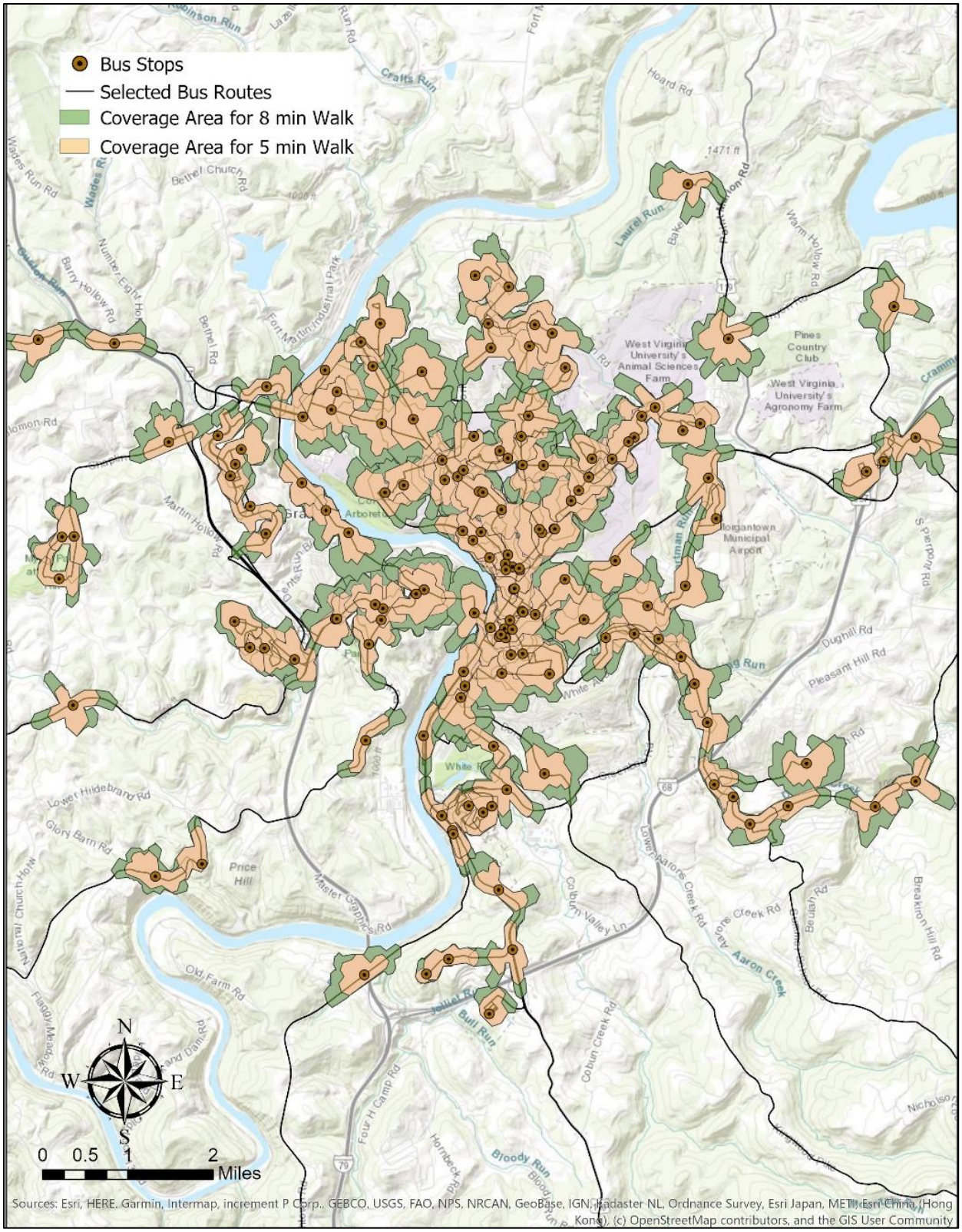


Figure 5.2: System-level improvement in transit stop service area coverage

Chapter 6: Conclusions and future research directions

This study investigated the benefits of a multimodal app that delivers RTI to the users in terms of three related performance measures- Stop Buffer Area Ratio (SAR), Coverage area improvement for the route, and Transit Supply Index (odSI). The first two measures are related to the spatial coverage of the transit route, and associated transit stops, whereas the third measure combines route connectivity and service availability to calculate the improvement in Origin-Destination-based available supply of the transit services. For this study, the multimodal app was developed on an android platform. The app was designed for a popular public transit route in Morgantown, WV, to quantify app benefits. Participants were recruited for field data collection with and without using the app (i.e., divided into control group-without app, and intervention group- with app). The control group performed trips from the defined Origin zones and selected Destination zones without using the multimodal APP. The control group trips were made only using the set bus schedule information. The users in the intervention group performed trips from the same Origin zones and went to the same Destination zones using the app. Transit stop wait time, travel speed, and travel time data were calculated using the trip's GPS trajectory via the multimodal app. Several previous studies found that RTI sharing among users can decrease stop wait time by conducting a survey or simulation-based analysis. This study utilized actual field trips data to measure the benefit and found that RTI can reduce the transit stop wait time by three minutes. This wait time saving was applied to calculate increased buffer area for each transit stop, increase in route level service area. With the app, the transit route service area can be increased by 37%. An Origin-Destination-based supply index (odSI) shows that the multimodal app increased transit supply compared to the scenario without the multimodal app. The increase in odSI value indicates that the transit system can be more sustainable when RTI of transit services were disseminated to users.

Although this study quantified the benefit of RTI sharing, several limitations could be addressed in a future study. In this study, the app was implemented to collect data for one route only. In the real world, there could be several routes that serve one Origin area. All transit routes could be studied to create a complete picture of the network-level benefits. Also, this study used walking mode only for the first- and last-mile connectivity. However, the app can significantly impact service area improvement when bicycles are used for first- and last-mile connectivity.

Future research could explore the impacts of both walking and bicycle modes as first and last-mile connectivity. The study was conducted in an area that is not densely populated. Future research should deploy the app in a more populated area to explore its effectiveness in an urban environment. Finally, this study did not consider intrazonal trips to calculate the odSI values; hence future research could include intrazonal trips in calculating the odSI matrix.

References

1. Miller P, de Barros AG, Kattan L, Wirasinghe SC. Public transportation and sustainability: A review. KSCE Journal of Civil Engineering. 2016 Apr 1;20(3):1076-83.
2. Bree S, Fuller D, Diab E. Access to transit? Validating local transit accessibility measures using transit ridership. Transportation Research Part A: Policy and Practice. 2020 Nov 1;141:430-42.
3. Zhang G, Yang H, Li S, Wen Y, Li Y, Liu F. What is the best catchment area of bike share stop? A study based on Divvy system in Chicago, USA. In2019 5th International Conference on Transportation Information and Safety (ICTIS) 2019 Jul 14 (pp. 1226-1232). IEEE.
4. Zuo T, Wei H, Rohne A. Determining transit service coverage by non-motorized accessibility to transit: Case study of applying GPS data in Cincinnati metropolitan area. Journal of Transport Geography. 2018 Feb 1;67:1-1.
5. Zuo T, Wei H, Chen N. Promote transit via hardening first-and-last-mile accessibility: Learned from modeling commuters' transit use. Transportation Research Part D: Transport and Environment. 2020 Sep 1;86:102446.
6. El-Geneidy A, Grimsrud M, Wasfi R, Tétreault P, Surprenant-Legault J. New evidence on walking distances to transit stops: Identifying redundancies and gaps using variable service areas. Transportation. 2014 Jan;41(1):193-210.
7. Foda MA, Osman AO. Using GIS for measuring transit stop accessibility considering actual pedestrian road network. Journal of Public Transportation. 2010;13(4):2.
8. Kim J, Kim J, JUN M, KHO S. Determination of a bus service coverage area reflecting passenger attributes. Journal of the Eastern Asia Society for Transportation Studies. 2005;6:529-43.
9. Murray AT, Davis R. Equity in regional service provision. Journal of Regional Science. 2001 Nov;41(4):557-600.
10. Brakewood C, Macfarlane GS, Watkins K. The impact of real-time information on bus ridership in New York City. Transportation Research Part C: Emerging Technologies. 2015 Apr 1;53:59-75.
11. Mostafizi A, Dong S, Wang H. Percolation phenomenon in connected vehicle network through a multi-agent approach: Mobility benefits and market penetration. Transportation Research Part C: Emerging Technologies. 2017 Dec 1;85:312-33.
12. Ubiergo GA, Jin WL. Mobility and environment improvement of signalized networks through Vehicle-to-Infrastructure (V2I) communications. Transportation Research Part C: Emerging Technologies. 2016 Jul 1;68:70-82.
13. Luo W, Wang F. Measures of spatial accessibility to health care in a GIS environment: synthesis and a case study in the Chicago region. Environment and planning B: planning and design. 2003 Dec;30(6):865-84.
14. Morris JM, Dumble PL, Wigan MR. Accessibility indicators for transport planning. Transportation Research Part A: General. 1979 Apr 1;13(2):91-109.
15. Malekzadeh, Ali, and Edward Chung. "A review of transit accessibility models: Challenges in developing transit accessibility models." International journal of sustainable transportation 14.10 (2020): 733-748.
16. Lei TL, Church RL. Mapping transit-based access: integrating GIS, routes and schedules. International Journal of Geographical Information Science. 2010 Feb 1;24(2):283-304.

17. Mavoa S, Witten K, McCreanor T, O'sullivan D. GIS based destination accessibility via public transit and walking in Auckland, New Zealand. Journal of transport geography. 2012 Jan 1;20(1):15-22.
18. Horner MW, Murray AT. Spatial representation and scale impacts in transit service assessment. Environment and Planning B: Planning and Design. 2004 Oct;31(5):785-97.
19. Faghih-Imani A, Eluru N. Analysing bicycle-sharing system user destination choice preferences: Chicago's Divvy system. Journal of transport geography. 2015 Apr 1;44:53-64.
20. Noland RB, Smart MJ, Guo Z. Bikeshare trip generation in New York city. Transportation Research Part A: Policy and Practice. 2016 Dec 1;94:164-81.
21. Ma T, Liu C, Erdoğan S. Bicycle sharing and public transit: does Capital Bikeshare affect Metrorail ridership in Washington, DC?. Transportation research record. 2015 Jan;2534(1):1-9.
22. Guerra E, Cervero R, Tischler D. Half-mile circle: Does it best represent transit station catchments?. Transportation Research Record. 2012 Jan;2276(1):101-9.
23. Ashraf MT, Hossen MA, Dey K, El-Dabaja S, Aljeri M, Naik B. Impacts of Bike Sharing Program on Subway Ridership in New York City. Transportation Research Record. 2021:03611981211004980.
24. Flamm BJ, Rivasplata CR. Public transit catchment areas: The curious case of cycle-transit users. Transportation Research Record. 2014 Jan;2419(1):101-8.
25. Galpern P, Ladle A, Uribe FA, Sandalack B, Doyle-Baker P. Assessing urban connectivity using volunteered mobile phone GPS locations. Applied Geography. 2018 Apr 1;93:37-46.
26. Cai, Z., Wang, D. and Chen, X.M., 2017. A novel trip coverage index for transit accessibility assessment using mobile phone data. Journal of Advanced Transportation, 2017.
27. Welch TF, Mishra S. A measure of equity for public transit connectivity. Journal of Transport Geography. 2013 Dec 1;33:29-41.
28. Zhao F, Chow LF, Li MT, Ubaka I, Gan A. Forecasting transit walk accessibility: Regression model alternative to buffer method. Transportation Research Record. 2003;1835(1):34-41.
29. Dey KC, Mishra A, Chowdhury M. Potential of intelligent transportation systems in mitigating adverse weather impacts on road mobility: A review. IEEE Transactions on Intelligent Transportation Systems. 2014 Dec 18;16(3):1107-19.
30. Brakewood C, Watkins K. A literature review of the passenger benefits of real-time transit information. Transport Reviews. 2019 May 4;39(3):327-56.
31. Zargayouna M, Othman A, Scemama G, Zeddini B. Impact of travelers information level on disturbed transit networks: a multiagent simulation. In 2015 IEEE 18th International Conference on Intelligent Transportation Systems 2015 Sep 15 (pp. 2889-2894). IEEE.
32. Gooze A, Watkins KE, Borning A. Benefits of real-time transit information and impacts of data accuracy on rider experience. Transportation Research Record. 2013 Jan;2351(1):95-103.
33. Papangelis K, Nelson JD, Sripada S, Beecroft M. The effects of mobile real-time information on rural passengers. Transportation Planning and Technology. 2016 Jan 2;39(1):97-114.
34. Liu Y, Shi J, Jian M. Understanding visitors' responses to intelligent transportation system in a tourist city with a mixed ranked logit model. Journal of Advanced Transportation. 2017 Jan 1;2017.
35. Cats O, Gkioulou Z. Modeling the impacts of public transport reliability and travel information on passengers' waiting-time uncertainty. EURO Journal on Transportation and Logistics. 2017 Sep 1;6(3):247-70.

36. Fries R, Dunning A, Chowdhury M. University traveler value of potential real-time transit information. 2011.
37. Brakewood C, Barbeau S, Watkins K. An experiment evaluating the impacts of real-time transit information on bus riders in Tampa, Florida. Transportation Research Part A: Policy and Practice. 2014 Nov 1;69:409-22.
38. Tang L, Thakuriah P. Will psychological effects of real-time transit information systems lead to ridership gain?. Transportation research record. 2011;2216(1):67-74.
39. Tang L, Thakuriah PV. Ridership effects of real-time bus information system: A case study in the City of Chicago. Transportation Research Part C: Emerging Technologies. 2012 Jun 1;22:146-61.
40. Hussain E, Bhaskar A, Chung E. A novel origin destination based transit supply index: Exploiting the opportunities with big transit data. Journal of Transport Geography. 2021 May 1;93:103040.
41. Fan Y, Chen Q, Liao CF, Douma F. Smartphone-based travel experience sampling and behavior intervention among young adults.
42. Guest G, Bunce A, Johnson L. How many interviews are enough? An experiment with data saturation and variability. Field methods. 2006 Feb;18(1):59-82.
43. Dziekan K, Vermeulen A. Psychological effects of and design preferences for real-time information displays. Journal of Public Transportation. 2006;9(1):1
44. El-Geneidy A, Grimsrud M, Wasfi R, Tétreault P, Surprenant-Legault J. New evidence on walking distances to transit stops: Identifying redundancies and gaps using variable service areas. Transportation. 2014 Jan;41(1):193-210.
45. MLTA Route list. [online] Available at: <https://www.busride.org/Maps-Schedules> [Accessed 17 June 2021].

An article based on this project has been submitted for presentation on Transportation Research Board (TRB) Annual Meeting 20212, and publication in TRB Transportation Research Record.

Appendix A: User Guide for the Multimodal App

Introduction

The overall goal of the project was to design a multimodal tracking app that tracked buses, the user, provided weather information, and travel advice to the user via a map. The app was designed to facilitate multimodal travel around a metro area, specifically Morgantown. In Figure 3.1 of the Method Section, the general app architecture was described. The following description further elucidates the figure:

The user device (e.g., an android smartphone, or other android device with cellular connection and a GPS) marked its own location and records its own speed on the app. It also periodically made calls to the Open Weather API, every two minutes, and the Bus Server, every five seconds, to receive data. The Bus Server was used to store the database.

A long click on the map (i.e., holding a finger on a specific place) causes the user device to send the coordinates of the click to Google Directions API. Google Directions API returns the directions to the coordinates, as well as the time and distance to the coordinates. The directions are highlighted on the map and the time and distance are displayed to the user. For more information, see the User Guide provided in the next section. The buses send their location to the server via the bus app, which must be present on the bus, and the server subsequently stores the data in the database, the last coordinate of which is sent to the client on request. The app presents the information in a map, provides weather information, bus information in Morgantown, WV, a select number of bus stops, directions in the form of a polyline, and distance and time to a destination. It also provides the functionality to choose the travel mode of the user and change the server for the buses.

For general information, including installation instructions and the functionality of the app, see the user guide below. Links are provided where appropriate. For developers looking to change or expand this app, can see the project code section, where a description is provided of the various files, classes, functions, and ideas. This section is meant as a helper for developers and should be used in conjunction with the code itself and the comments therein.

Installation of the Required Components:

Server:

1. Install tomcat on the server. See the Tomcat Website itself for this: http://tomcat.apache.org/tomcat-9.0-doc/index.html. The version of Tomcat that the system is verified to work with is apache-tomcat-9.0.38.
2. Download the WAR file in the dist folder of the server. Available at:
    a. https://drive.google.com/drive/u/0/folders/10j_EKFXZS9IiLIwTfzM6bg8wG1hOHXIB
    b. OR https://github.com/AnCarro/MultiModalServer
3. Move the WAR file into the webapps folder of Tomcat, and start Tomcat, to start the server. NOTE: Before the server will run properly, the Database must be installed. Described below in step four.

a. To start Tomcat, navigate to the bin folder inside the Tomcat folder. For windows, double-click **startup.sh**. For Linux, double-click **startup.bat.**
b. Tomcat will then run and deploy the WAR file in the webapps folder. To verify, look in the webapps folder and see that the WAR file has been deployed into a folder named identically. I.e., in this case, WebApplication1.

4. Install MySQL. Download the Dumps folder provided at: https://github.com/AnCarro/MultiModalServer or the Dump20210510 folder provided at: https://drive.google.com/drive/u/0/folders/10j_EKFXZS9IiLIwTfzM6bg8wG1hOHXIB
5. Import the Dumps folder to the database.
    a. This is easily done using MySQL Workbench.
        i. Server -> Data Import.
        ii. Select the 'Import from Dump Project Folder' button and navigate to the dumps folder.
        iii. NOTE: Depending on the project intentions, it may be desirable to clear the tables as there is test data currently stored. To do this, run this query: TRUNCATE TABLE tablename; where 'tablename' is the name of the table to be clear.

User app:

The user interface of the user app is shown in Figure A.1.

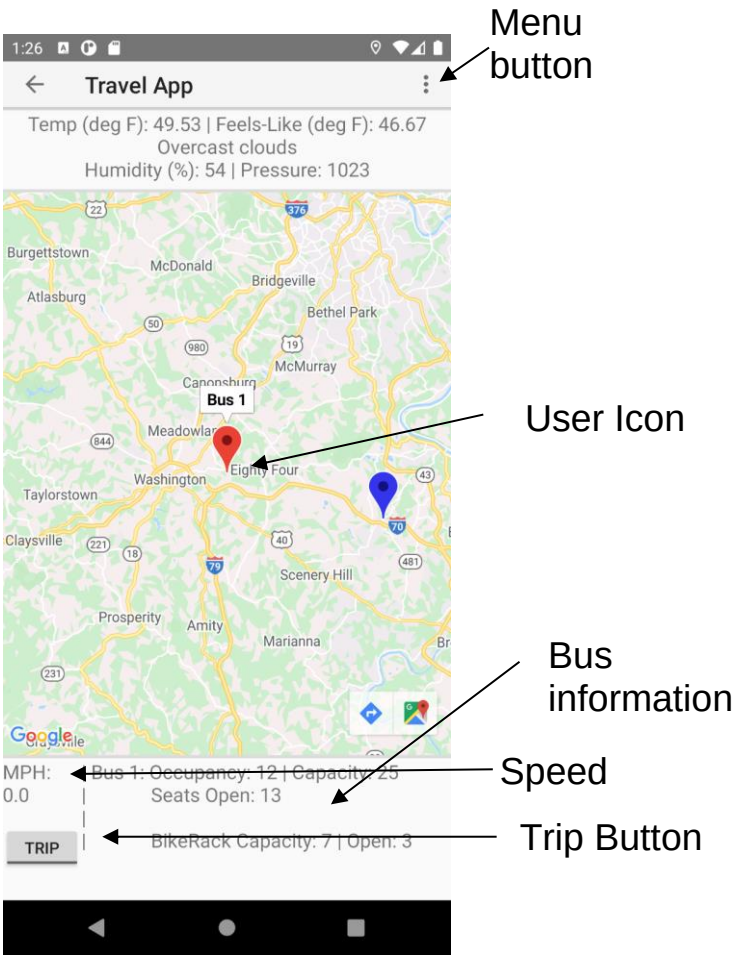


Figure A.1. User app main page

Installation:

NOTE: This app only works on Android devices.

To install, find the APK at:

https://drive.google.com/drive/u/0/folders/10j_EKFXZS9IiLIwTfzM6bg8wG1hOHXIB

Or on the github at:

https://github.com/AnCarro/MultiModalUserSideApp

NOTE: Due to the sensitive nature of the API key, the user must be given permission to access the github. Request this from the developer.

There are two choices from here. The APK can download it onto a computer and then moved the device via USB cable. Or the link could be accessed via phone and the APK downloaded directly. This article is helpful for either action:

https://www.lifewire.com/install-apk-on-android-4177185

Note that the easiest way to the author is to download the APK onto the computer, connect the phone, after enabling USB debugging (https://www.lifewire.com/enable-usb-debugging-android-4690927) finding the APK in files, and installing it.

The user app consists of three pages. The first page, the Map Page, is the most important: It contains the majority of the information, including the map, the user's location, the weather, the bus information (e.g., location, available seating and bike rack capacity), and the bus stop information. The other two pages consist of the Server page, and the Travel Info page. All three will be explained in more detail below.

Upon loading the app for the first time, it will request access to the location of the device. They should agree, so that it can track the real-time location. If the app does not have access to the location, it may appear in the middle of the ocean. To fix this, make sure that location is turned on and the app has permissions to access the location. Then reload the app. From here, the app should move the user to their current location.

Setting the Server:

To receive the bus information, the user must enter the proper server information, from the developer or administrator of the server. To set the server, select the three periods in the top right-hand corner of the screen (See figure A.1), on the toolbar. Select the server option as seen from figure A.2.

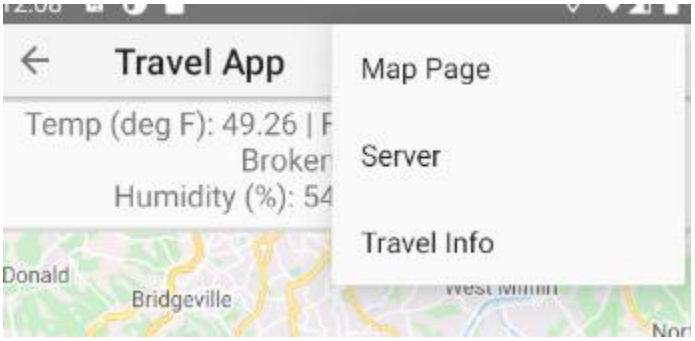


Figure A.2: Setting the server.

Type in the appropriate server and select change (Figure A.3). As of this writing, the server should be entered as:

http://hwsrv-779048.hostwindsdns.com:8080/

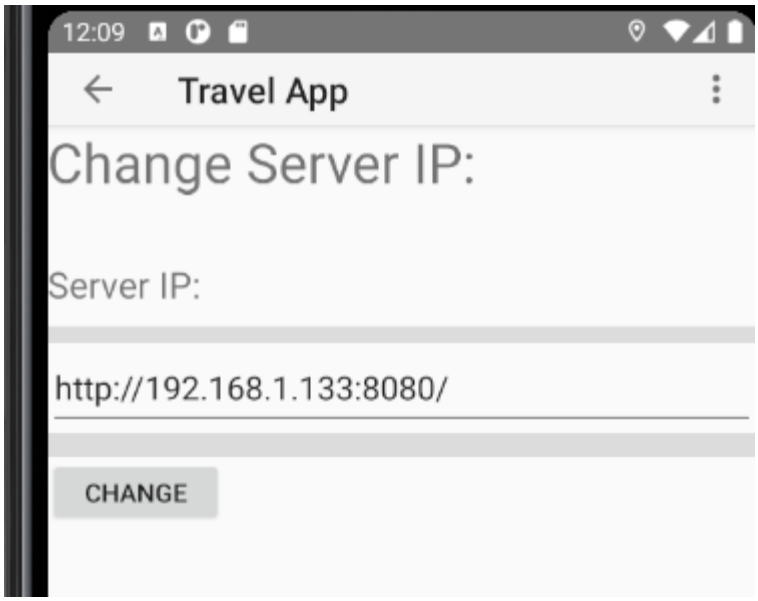


Figure A.3: Server IP address.

If the buses still do not appear, the most likely explanation would be that the server is shut down. Turn it on (Steps are shown in Server Maintenance section) and make sure that there are no errors. If there are, close out of the Tomcat window, and restart it.

Note that the app does not have the capability to remove inactive buses, so this information should always be there if the server is running and has not crashed. To remove buses, see the discussion about clearing the tables, under Installation.

Using the App:

Trips:

The app records each trip to draw general information from the recorded data for future use. To start a Trip:

- Click the 'Trip' button beneath the speedometer.
- The user will be taken to a new page, where the user must input his/her name.
- Once a first and last name are input, select the 'Start Trip' button.
- A trip number will automatically be assigned and displayed underneath the button (The 'Start Trip' button will have changed to 'STOP'.
- Note the trip number
- When the user has finished the trip, select stop. The user will be automatically directed to a survey URL.
- Input the trip number at the top, and then continue the survey as directed. If the user has forgotten his/hertrip number, it should be saved in the clipboard. Simply paste it in.

Other Data:

There are several other functionalities that may not at first be apparent.

- By selecting a bus stop marker (only in Morgantown on the selected route) the estimated time for arrival for a bus will pop up. This information is based on the available bus schedule for the selected route.
- The user can press and hold their finger on a location on the map, and the app will calculate the time and distance it would take to reach that spot (given for walking, but this can be changed, as will be explained next)
- The user can tap on the bus itself to receive the buses information. The information will appear on the bottom window, detailing the occupancy and capacity information, and the bike rack information, along with the real time speed.
- The user can change his/her travel mode. That is under the Travel Info tab on the menu. Simply tap it and select the preferred travel method. The user can also change how the units are displayed here.
- Finally, the weather appears at the top of the screen. It updates periodically (every two minutes) and is only for Morgantown.

Troubleshooting:

If the app stops working at some point, there are a few fixes to attempt, though only a few are available to the user him/herself. Below the possible fixes are given, along with the necessary access level.

- Make sure that the server is still running wherever it is installed. At this time, it is: hwsrv-779048.hostwindsdns.com
  - Access Level: Admin/Developer
- If the server is still running, check to see errors in the Tomcat window. If there are some, note what caused the errors, copy the errors from the window, and contact the developer, sending him/her the errors. Then restart the server and continue using the app.
  - Access Level: Admin/Developer
- Restart the app. Make certain that the app is running the correct server.
  - Access Level: Anyone
- Restart the server. Make sure it loads the WAR file properly.
  - Access Level: Anyone
- Replace the WAR file and delete the folder created when the WAR file is deployed. Restart Tomcat and make sure the folder is redeployed. Note any errors.
  - Access Level: Admin/Developer.

BusApp:

A layout of the bus travel app is given in Figure A.4.

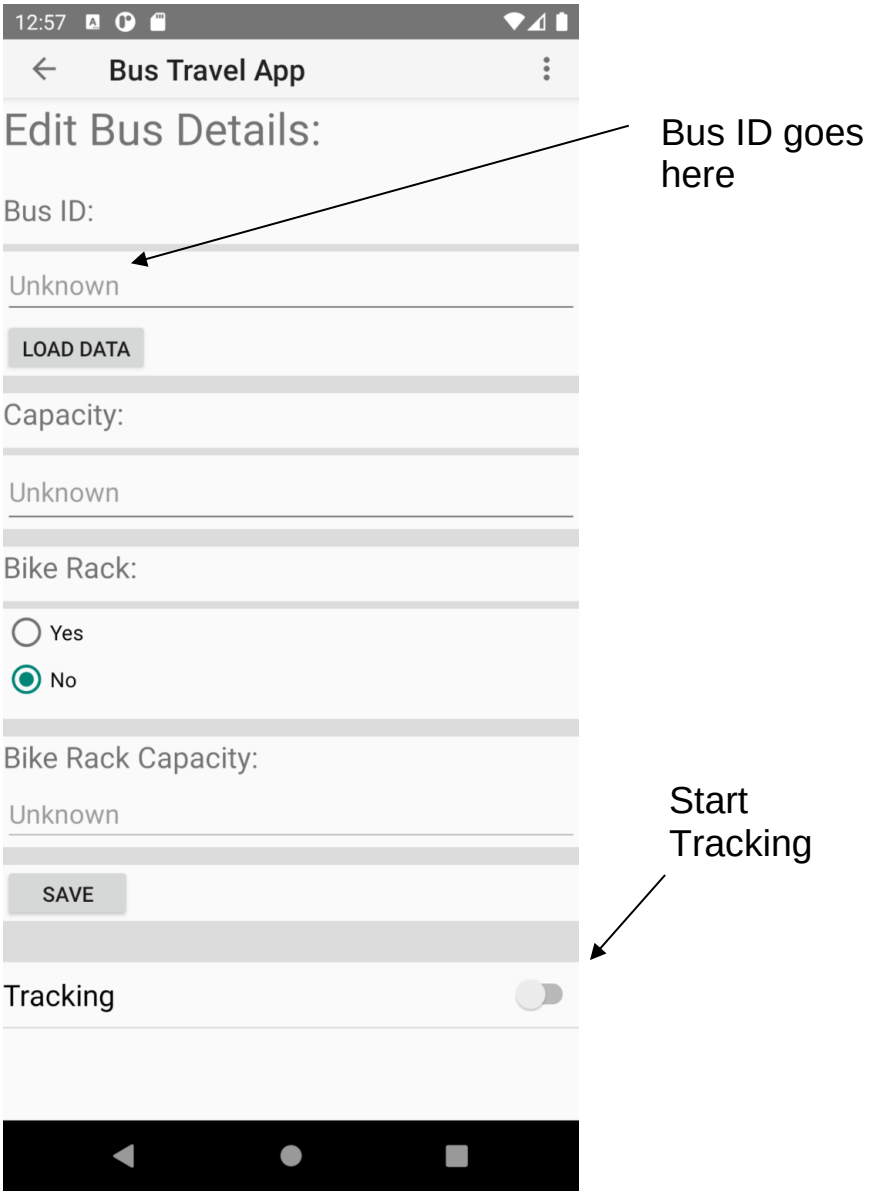


Figure A.4. Static Bus Info Page

Installation:

Follow the same instructions as those laid out for the User App. The APK is given by the same google link. However, for the github to acquire the APK, use the below link:

https://github.com/AnCarro/MultiModalBusTravelApp

NOTE: This app only works on Android devices.

The BusApp's main use is the tracking and detailing of buses. To accomplish this, the app employs three pages.

- Static Bus Info - Contains the static bus information: the bus ID, bike rack information, and capacity.
- Dynamic Bus Info - Contains the dynamic bus information. That is the occupancy and bike rack occupancy.
- Server - Contains the server that the app is connected to.

The first step using the Bus Travel App is to establish connection to the server. To do this, navigate to the three buttons on the top right of the screen and select the Server option. After that, input the following server URL: http://hwsrv-779048.hostwindsdns.com:8080/

The next important step is to enter the Bus ID on the Static Bus Info page.

Input the Bus ID. If the user believes the information for the bus already exists, select 'LOAD DATA'. If the bus exists and the app has access to the proper server, the data will load.

If the data does not load, the bus by the given ID does not exist and the user will need to input the data manually. If the loaded data is incorrect, input the new values and select the 'SAVE' button to store it.

NOTE: There is a known bug in this process, in that the data loads improperly at first. To fix this, simply navigate to another page and return. The data should be fixed.

The final important step is to begin tracking. This option is at the bottom of the Static Bus Info Page. See figure A.4 above. Simply turn the option on to begin tracking.

Project Code

Below is a description of the project code and its various parts, beginning with the server. This section is meant for any future software developers, to make the code and ideas behind the project as simple to understand as possible.

Server:
The server is built using Apache Tomcat, an open-source Java application created by Apache. It consists of multiple classes.

Classes:

Bus.java

Encapsulates a bus for the server.

Contains multiple constructors, accepting an ID and/or coordinates in various forms.

Coordinate.java

Encapsulates coordinates, latitude and longitude and speed.

Multiple constructors accepting various forms of latitude and longitude and speed

BusInfo.java

Encapsulates the bus information, including a value for the capacity, occupancy, seats left, percent full, bike rack capacity, bike rack occupancy, and the number of bike racks left.

The seats left, bike racks left, and percent full are all calculated within the class when an object is made.

DBManager.java

Encapsulates the database. The constructor sets up the database

The functions are as follows:

addNewCoordinatesToMode (PK, lat, lng)

Adds coordinates to a given mode (bus) using the PK to determine which mode, and the lat and lng as the coordinates.

addNewCoordinatesToMode (PK, lat, lng, speed)

Adds coordinates to a given mode (bus) using the PK to determine which mode, and the lat and lng and speed as the coordinates.

replaceStaticBusData(PK, capacity, bikeRackCapacity)

Adds information about thebus to the bus_info table. The PK determines which bus.

insertDynamicBusData(PK, occupancy, bikeRackOccupancy)

Inserts dynamic data into bus_dynamic_info, using PK as the bus.

newBusOccupancy(PK, occupancy)

Changes out the occupancy.

addNewUser(usern, vehicle, pswd)

Adds a new user. Currently not needed.

testUserLogin(username, password)

Tests if a user is logged in. Currently not needed.

Coordinate returnLastCoordinate(PK)

Returns the last coordinate for the vehicle with the matching PK, as a Coordinate Object.

VehicleInfo getVehicleInfo(PK)

Returns the vehicle information for the given PK. The information is returned as a Vehicle Info object, which contains the name of the vehicle and the type of it.

List<String>getTransportPKs()

Returns a list of the PKs for all active transport units.

returnBusInfo(PK)

Returns bus information for the bus specified by the PK, which is the capacity, occupancy, bike rack capacity, and bike rack occupancy. If one is empty, it will show up as a negative 1.

GetDataSource.java

Encapsulates the data source of the database. Used in the DBManager class to get the database data source.

Servlets:

CheckForData

Checks the Database for data on the active buses. If there is data, it sends back all the buses as a JSON Object.

ReceiveData

Receives data stores it in the server. The data it receives are:

- The PK -- the Primary key for the bus, which is used to identify the different buses.
- The latitude shortened to the lat.
- The longitude which is not shortened.
- The time. The time is currently not used or stored in the database. It may become useful later, however.

GetBusData.java

On call, checks to get a string called "PK". Once attained, it queries the database for the bus info, see the return 'returnBusInfo' function in the DBManager app. The servlet then sends the BusData object back as JSON.

ReceiveDyanmicBusData

Gets the PK, the occupancy, and the bikeRackOccupancy from the client and sends it to the database, after error checking, through the insertDynamicBusData function (see DBManager class)

ReceiveStaticBusData

Gets the PK, the capacity, and the bikeRackCapacity from the client and sends it to the database, after error checking, through the replaceStaticBusData function (see DBManager class)

Client - User - Android App:

MapsActivity.java

The main class for the main activity of the Android App. This page includes the map and two text views for the weather and distance/location information. It is responsible for scheduling periodic calls to the server (using the ServerRequests class), periodic calls to the weather server, periodic location updates, and making the calls to the DirectionsAPI service on google's servers. A description of the functions follows:

Global Variables

- GoogleMap mMap
  Responsible for holding the map object to be referenced throughout the lifetime of the app
- static User
  User is an object to encapsulate the user. See User class
- VelocityConverter Converter
  The object to encapsulate the converter from one velocity measurement to another. Also contains the enums for the different velocity measurements

- Context context
  Holds the context for the app to be referenced when needed
- FusedLocationProviderClient FusedLocationClient
- LocationRequest LocationRequest
- LatLng CurLoc
  Stores the current location of the user
- RequestQueue rQueue
  Holds the RequestQueue for servlet requests
- TextView TopTextView
  The TextView for the top of the main page of the app
- TextView BottomTextView
  Bottom text view of the main page of the app
- TripNumberTextView
  Contains the trip number when a trip is active
- SupportMapFragment mapFragMent
- Polyline CurrentHighlightedRoute
  Holds the currently highlighted route for the directions
- static String TravelMode
  Holds the travel mode which the app uses to calculate distance/directions
- static String ServerURL
  Holds the URL the app makes calls to for bus data

onCreate(Bundle savedInstanceState)

The function that is called when the activity is being created.

Responsible for starting up most of the parts of the app. Sets the ToolBar, calls callAysnchronousTask(). Checks/requests location permissions. Acquires phone locations, and begins call back for continuing to acquire location and speed.

Note the TripButton for setting or turning off trips. When checked, it class the enterTripInfo() function. When checked off, calls the afterTripActivity() and clears the TRipNumberTextView.

onCreateOptionsMenu(Menu menu)

Adds options menu

onOptionsItemSelected(MenuItem item)

On menu selection, call appropriate functions to act appropriately.

changeTravelMode()

Called in onOptionsItemSelected(), this function changes to the ChangeTravelMode activity.

changeServerURL()

Called in onOptionsItemSelected(), this function changes to the ChangeServerURL activity.

onMapReady(GoogleMap googleMap)

When the map is ready, sets the mMap global variable. Also sets the listeners on the map, adds the user location, and the buses, as long as they are not empty.

onMapLongclick(LatLng point)

Listener for a long click. Calls DistAndDirections passing it the point clicked, and the travel mode.

onMapClick(LatLng point)

Listener regular click. Clears the polyline if the polyline is on the map.

requestPermissions()

Requests necessary permission for location access

callAsynchronousTask()

Calls our two asynchronous tasks, which are the doServerCall (for bus data) and the doWeatherCall (for weather data). Sets both tasks on asynchronous timers.

startLocationUpdates()

Starts location updates callback/periodic updates. Also provides speed

requestNewLocationData()

Requests a new location update

LocationCallBack()

addStops()

Function to add stops to map

DistAndDirectionsRenderer(LatLng dest, String mode)

This function first builds the request to send to the Google API for directions to a certain destination (dest) using the given mode of transportation (mode). Once it receives the request, it parses through the data, marking the time and distance on the bottom Text View. It also parses out the encoded polylines and marks them on the map, saving them in CurrentHighlightedRoute.

onMarkerClick(Marker marker)

Sets the arrival times if a bus stop has been clicked, or sets the bus information if a bus marker was clicked

set____ArrivalTime (Marker marker)

Where the blank space is a given bus stop and the marker is a given bus marker selected. Based on the offset, uses the returnETA to give us the ETA of a bus to the selected marker.

returnETA(int offset)

Calculates the current time in minutes, then uses the mathematics and modular division to determine the buses nearest arrival time. Sets the string and returns it.

setBusDetails(sID)

Where sID is the bus ID, Queries the database using the GetBusData servlet to get the bus information, and then sets it in the BottomTextView of the User app

showSnackBar(message)

Shows a snackbar with the given message. Is currently not used

enterTripActivity()

Puts user into the activity that starts the trip

afterTripActivity()

Puts user into the activity that ends the trip

ChangeTravelModeActivity.java

This is the class for the activity that allows the user to change the travel mode between, walking, biking, and driving.

ChangeServerUrlActivity.java

This is the class for the activity that allows the user to change the server.

TripDetailsActivity.java

This class is responsible for setting the trip activities. It contains the TripId and the TripStartTime, both which are set in this activity. The first name and last name must be entered in order for the trip to begin.

AfterTripActivity.java

This class is responsible for displaying the TripId and length of the trip to the user, as well as redirecting them to the encoded survey URL when they select the button

User.java

This class encapsulates the user and all of their actions

- LatLng CurrentLocation-- The current location of the user
- LatLng PreviousLocation-- the previous location of the user
- Float CurrentVelocity -- the current velocity of the user
- float PreviousVelocity -- the previous velocity of the user
- Marker Marker -- the marker representing the user's location
- GoogleMap Map -- the internal map sent from the main project and used to set all the markers
- String Title -- the title for the user's marker
- Velocity Converter Converter -- the converter for the User class
- VelocityConverter.Units Units -- the velocity units the user is currently using
- String StringUnits -- the string version of the user's units

The class contains multiple constructors. Each one sets the global variables and the marker for the bus. Units are not required to create the bus, and if they are not provided, they will be defaulted to MPH.

Other functions:

Move (LatLng newLoc)

Moves the bus if the marker exists. Sets PreviousLocation to CurrentLocation and CurrentLocation to newLoc;

refreshMarker(GoogleMap map)

Refreshes the google maps marker, along with the map marker, if the map has changed (which it will on reload of page)

setVelocity(float vel)

Sets PreviousVelocity to CurrentVelocity. Sets CurrentVelocity to vel, after Vel is set to the converter, with Units.

setUserUnits(VelocityConverter.Units units)

Sets Units to units. Converts units to a string and sets StringUnits.

setUserUnits(String units)

Identical to previous function, except that units comes in as a string and must be appropriately converted.

These functions are followed by several getters which are obvious enough upon inspection.

Bus.java

The class that encapsulates the bus data. Contains multiple constructors for buses to set the global variables, which are:\

- String id -- The id of the bus
- LatLng location -- the current location of thebus
- LatLng prevLocation -- the previous location of thebus, if there was one
- Marker marker -- the marker object of the bus
- String title -- the title of the bus
- GoogleMap Map -- the map sent from the main program

Also contains two separate functions:

Move(LatLng newLoc)

Takes in the new location and sets the new position for the marker.

refreshMarker (GoogleMap map)

Refreshes the marker, if, somehow, it would get deleted. (Generally, on reload of the MapsActivity)

ServerRequest.java

The class that contains the information to make periodic requests to the server for bus information. In order to be periodic, it has to extend AsyncTask<Void, Void, String>

doInBackGround(Void...voids)

The task which can be scheduled and therefore calls the SetBusPostions();

SetBusPositions()

The class's main function which calls to the server, and gets the bus data, which it parses, and places into the Static MapsActivity.Buses hashmap.

VelocityConverter.java

The class to take care of velocity unit conversions. Contains an enum to represent the various units. As of this writing, these are: MPH, KMPH, MPS, FPS. There are no other global variables.

The constructor requires no inputs, as there are no global variables.

There are several conversion functions converting MPS to the above units. These functions are duplicated, oftentimes with the difference of accepting a string instead of a float.

Due to the fact that the getSpeed function inherent in the location object (see MainActivity.java) returns MPS, there is no need to implement any conversion aside from MPS to the other units.

The MPS conversion functions are private and are called in:

float ConvertMPSTo()

There are multiple ConvertMPSTo functions. They accept a unitType to convert to and a velocity to convert.

These functions accept both Units (from the enum) and Strings as the units, and floats and strings as the velocity. They then take care of returning the proper converted float, as well as rounding it (see the round function).

String unitsToString(Units unit)

Converts Unit to string

Unit StringToUnits(String sUnits)

Converts sUnits to Units

float round(float value, int precision)

Rounds value to the precision specified by precision.

WeatherServerRequests

Nearly identical to the ServerRequest.java except that it calls to the weather server, and instead of putting the data into the buses hashmap, puts the parsed weather data into the Top TextView

ServerRequestsQueue:

Class to encapsulate the Request Queue for the voley calls.

activity_maps.xml
The UI design for the main page activity.

change_server_url.xml
The UI design for the Change Server Activity.

change_travel_mode.xml
The UI design for the Change Travel Mode Activity.

travelapp_menu.xml
The menu for the Travel App

trip_activity,xml
The design for the initial trip display page.

after_trip_activity.xml
The design for the page that displays after the rip is over.

Client - Bus - Android app

This is the app for the bus. It is responsible for sending the bus information, including latitude and longitude, occupancy/capacity information, and bike rack information to the server. It allows the user to change all of the bus data and updates the database with the changes.

StaticBusInfoActivity

The page responsible for setting the static bus information. The user (e.g., bus operator) can enter static bus information, including the PK, the capacity of the bus and the bikerakc, to update it or send new records to the database.

onCreate(Bundle savedInstanceState)

This is the main function on this page. It is responsible for all of the text input, button pressing, most of the user interaction.

To keep this function easier to read, every button or EditText is put in its own scope.

There are multiple other functions on this page not explained here. Most of them have to do with OnClick functions, sending data to the server, or accessing the phone location.

DynamicBusInfoActivity
The page responsible for setting the dynamic bus information

ChangeServerUrlActivity
Page responsible for setting the server's location

static_bus_info.xml
Layout for the static bus information page.

dynamic_bus_info.xml
Layout for the dynamic bus information page.

change_server_url.xml
Layout for the change the server page.

travelapp_manu.xml
Layout for the menu, taken from the user app.