A Vehicle-based Edge Computing Platform for Transit and Human Mobility Analytics

Bozhao Qi
University of Wisconsin-Madison
bqi2@wisc.edu

Lei Kang
University of Wisconsin-Madison
lkang@cs.wisc.edu

Suman Banerjee
University of Wisconsin-Madison
suman@cs.wisc.edu

ABSTRACT

This paper introduces Trellis — a low-cost Wi-Fi-based in vehicle monitoring and tracking system that can passively observe mobile devices and provide various analytics about people both within and outside a vehicle which can lead to interesting population insights at a city scale. Our system runs on a vehicle-based edge computing platform and is a complementary mechanism which allows operators to collect various information, such as original-destination stations popular among passengers, occupancy of vehicles, pedestrian activity trends, and more. To conduct most of our analytics, we develop simple but effective algorithms that determine which device is actually inside (or outside) of a vehicle by leveraging some contextual information. While our current system does not provide accurate actual numbers of passengers and pedestrians, we expect the relative numbers and general trends to be fairly useful from an analytics perspective.

We have deployed Trellis on a vehicle-based edge computing platform over a period of ten months, and have collected more than 30,000 miles of travel data spanning multiple bus routes. By combining our techniques, with bus schedule and weather information, we present a varied human mobility analysis across multiple aspects — activity trends of passengers in transit systems; trends of pedestrians on city streets; and how external factors, e.g., temperature and weather, impact human outdoor activities. These observations demonstrate the usefulness of Trellis in proposed settings.

CCS CONCEPTS

• **Information systems** → *Location based services*;

KEYWORDS

Edge computing, In-vehicle system, Mobile computing, Transit analytics, Human mobility

ACM Reference format:
Bozhao Qi, Lei Kang, and Suman Banerjee. 2017. A Vehicle-based Edge Computing Platform for Transit and Human Mobility Analytics. In *Proceedings of SEC '17, San Jose / Silicon Valley, CA, USA, October 12–14, 2017,* 14 pages.
DOI: 10.1145/3132211.3134446

DOI: 10.1145/3132211.3134446

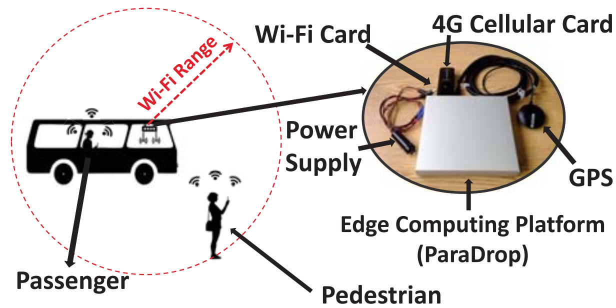


Figure 1: The on-board edge computing platform. The key challenge for Trellis is to determine whether an individual is located inside the vehicle (passenger) or outside of it (pedestrian). Based on the fact that pedestrians will eventually be out of monitoring range, Trellis solves this problem by observing device signal strength coupled with the vehicle's speed of movement.

1 INTRODUCTION

A public transit system is an important part of public infrastructure provided by local governments. According to the American Public Transportation Association's report [33], 10.6 billion public transportation trips were taken by Americans in 2015. An efficient and high quality public transportation system both benefits passengers and also has a large impact on city development. Hence, public transit operators have always looked for mechanisms that allow them to improve their services regarding issues such as what new routes or stops should be introduced, how peak and off-peak behaviors are handled, and much more.

Traditionally, these decisions are often based on limited surveys — metro transit operators would recruit volunteers and ask them about their experiences and transit preferences. However, just as mobile devices have transformed crowd-sourced data collection in a whole range of domains, we believe that transit systems can also benefit significantly from them. In this paper, we advocate a fairly low-cost and simple system through which a transit operator can gather significant user and usage analytics about its operations at a scale never possible before.

Transit systems need to learn much about transit usage to evaluate current transit routes/schedules, and to make decisions on adjustments [12]. Therefore, transit operators are actively seeking approaches that can answer questions such as: What are the most popular stops at different times of the day? How long do people at bus stops wait for the next vehicle? How occupied are different vehicles at different times of the day? What do public mobility patterns look like throughout a year, especially during hot summers and cold winters? Some of these questions are significantly related to funding allocations— in particular, operators sometimes receive government funds based on how many passenger-miles they carry annually [6, 19, 29]. Transit operators use a number of low fidelity methods to collect such information. However, existing solutions either failed to answer this question or have been too expensive to be widely deployed. For instance, most ticketing systems on metro buses can infer where passengers get on a bus, but they do not record where/when passengers get off a bus. Some public transit operators rely on expensive sensor systems to count the number of passengers as they get on and off the bus. But these systems are not able to detect the specific origin and destination of individual passengers. Camera-based solutions involve costly hardware and may generate privacy concerns when customers' facial identities are captured by the cameras. Even cameras are deployed, it is still very challenging to track individual passengers [13, 14]. What's more, pedestrian flows could eventually affect traffic conditions [23, 25–27]; however, there is not an effective method to estimate the number of pedestrians on the street. The approaches above tend to provide incomplete data or data with fairly low fidelity. In this paper, we propose a low-cost, wireless-based mechanism to conduct spatial-temporal public transit analytics and answer these unresolved questions.

The usage of edge computing platform: In-vehicle computing platforms are becoming increasingly important as they enable advanced safety, efficiency, and diverse services such as entertainment, navigation, and much more. Compared to cloud computing platforms, such computing platforms in the vehicles provide unique edge services with a lower latency, greater responsiveness, and more efficient use of network bandwidth. These characteristics create such in-vehicle computing platforms as unique locations in which edge computing can be effectively implemented. For example, the massive amount of data generated by the sensors on a self-driving vehicle needs to be processed in a timely manner. A vehicle-based edge computing platform would be an ideal place to execute these kinds of computing tasks. In our efforts, we use a previously developed edge computing platform — ParaDrop [30] — as the platform of choice for deployment inside vehicles to flexibly provide computing and storage resources, allowing developers to create various kinds of services.

We use the ParaDrop platform as a computing platform in vehicles because of its flexibility and its benefits in this environment. The ParaDrop platform, implemented on low-end Wi-Fi Access Points (APs), supports multi-tenancy and a cloud-based back-end through which computations can be orchestrated across many such APs. ParaDrop also provides APIs through which developers can manage their services across diverse ParaDrop APs. In this work, we installed a ParaDrop AP into a public transit vehicle in order to focus on our desired problem — conducting transit analytics. Various kinds of analytics can be done on this vehicle-based edge computing platform, such as video analysis and obstacle sensing. By loading computation tasks from the cloud to ParaDrop, our system achieves greater traffic efficiency while accomplishing our desired goals. Using ParaDrop, various relevant transit analytics can be quickly derived on-board and sent back to transit operators without incurring high data requirements from the vehicles. Additionally, it is easy to deploy and manage such applications in multiple vehicles across a whole city using ParaDrop.

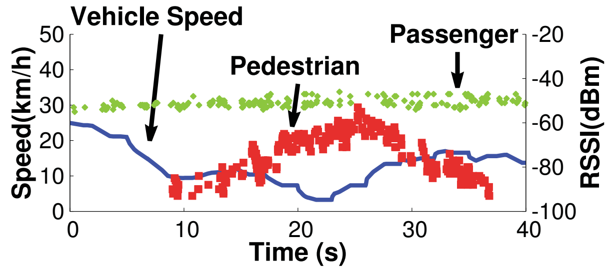


Figure 2: Different RSSI patterns between passenger and pedestrian.

A first look at Trellis: Wi-Fi-enabled mobile devices have skyrocketed in most parts of the world, and many reports point to their deep penetration among their populations [38]. Our proposed system, Trellis, takes advantage of these widely available mobile devices among passengers and pedestrians to quickly gather various forms of usage information at a significantly large (city) scale. A Wi-Fi-based monitoring system has been widely used in many related scenarios, such as tracking human queues [44], estimating vehicle trajectories [36], and understanding network performance [15]. Trellis uses this kind of mechanism in a similar but much simpler way. As shown in Figure 1, our system uses a low-end Wi-Fi monitoring unit mounted on the vehicle to distinguish passengers from pedestrians and determine when a certain passenger gets on and off the vehicle. The approach relies on the fact that many mobile devices typically have their Wi-Fi function turned on, which makes them trackable by another Wi-Fi observer. Most analytics in Trellis are based on the ability to distinguish between which individual is actually inside the vehicle and which is actually outside. While one may be tempted to utilize any one of a slew of Wi-Fi based localization techniques [11, 21, 22], the accuracies of these systems are often not sufficient to distinguish between a passenger seated inside the vehicle and a person who is just outside.

The approach in Trellis to make this distinction is fairly simple — when the vehicle is in motion, the signal strength of a passenger's Wi-Fi device as perceived by a vehicle-mounted Wi-Fi observer is likely to be fairly stable; while the signal strength of an outside pedestrian will vary in a predictable way before eventually disappearing (Figure 2). Thus, by simply observing the signal strength trends of Wi-Fi devices while a vehicle is in motion, this localization problem becomes quite simple and can be solved fairly accurately. This basic observation forms the core of many of our analytics presented in Trellis.

Obviously systems such as Trellis will not be able to count for passengers who travel without mobile devices or those with their Wi-Fi function turned off, but our observation shows that we can still track general trends in transit behavior quite effectively [1].

[1]Beginning with iOS 8, Apple introduced randomized MAC address techniques. Google also implemented similar techniques in Android version 6.0 or above. MAC randomization may lead to miscounting, but the trends still hold. We will discuss this effect more in section 8.1.2.

We recommend our current version of Trellis be used to track relative trends in transit systems, as opposed to using it for exact and absolute counts.

We believe that a simple and low-cost infrastructure such as Trellis mounted on public transit vehicles can be effectively used to perform transit analytics as well as answer questions regarding human mobility behavior studies. For the purpose of this work, we demonstrate how such a system may be used from three major perspectives. First, we focus on passenger riding habits, i.e. what are the origin-destination pairs of the user population and how does the popularity of these origin-destination pairs vary for different stations, at different locations, and at different times of the day. Next we study patterns of people on city streets: For example, how busy city streets are, and where hotspots are during different times of day and periods of the year. A system such as Trellis can provide such insights. Finally, we study the impact of weather on outdoor human mobility. Specifically, we observe how inclement weather (snow and rain) and outside temperature affects the number of people in transit vehicles or out on city streets.

In the end, Trellis provides a unique approach to collect transit information (in addition to other kinds of information) in real-time and can potentially be combined with other existing or complementary approaches. Overall, Trellis provides a new lens of human mobility at large scales. While we provide some initial aspects one can learn from this system, we believe many significant opportunities potentially exist.

Contributions. This chapter presents a low-cost in-vehicle wireless monitoring system that can track passenger movements and study pedestrian behaviors to assist transit operators, and potentially city planners, with various forms of human mobility analytics. We develop several simple heuristic algorithms that can effectively separate passengers from pedestrians and identify where passengers get on or off a vehicle. To test the efficacy of our system, we deployed Trellis on vehicle-based edge computing platforms over a period of ten months and collected data from 3 bus routes. We evaluate how it can be used to infer origin-destination pairs that are popular among passengers over time and space. We demonstrate and quantify different impacts on human activities caused by different factors (e.g., weather and temperature). As we continue to work with our local transit partners, we continue to evaluate how such a system can be used to identify where to add new bus routes, or when to add non-stop services between various stations throughout the city at different times of the day and under different weather conditions.

2 TRELLIS SYSTEM DESIGN AND IMPLEMENTATION

In this section, we discuss the overview, design, implementation and deployment of the system.

2.1 System Overview

Trellis tracks people by tracking their Wi-Fi-enabled devices. It achieves this goal in two steps. First, our system performs device detection tasks by capturing Wi-Fi transmissions from each device. As long as its Wi-Fi function is turned on, a device will send out probe request packages scanning for available access points. Our system takes advantage of this feature to capture Wi-Fi enabled devices. The system distinguishes devices by checking their MAC addresses. Once the system successfully detects a device, it will determine whether the device is inside or outside a vehicle. Our system determines the position of the device by observing RSSI coupled with the vehicle's speed of movement. After these two steps, the system records the device data into databases.

2.2 System Design

Our system uses a front-end monitoring module to collect Wi-Fi devices' signals and transit GPS information, and it uses a back-end processing module to reconstruct transit schedules and human mobility patterns. The monitoring module performs sniffing tasks and collects the data from mobile devices. The collected data will be saved in a local database along with corresponding GPS location information. Meanwhile, the sniffing module can send calculated passenger and pedestrian numbers to a remote server in real-time through a cellular link, i.e., for the purpose of real time monitoring. Although our system supports real-time communication, we use a separated program to send the data from the databases to a remote back-end server. The back-end server reconstructs public transit schedules and human mobility patterns from the collected data. It further combines the data from multiple transit sniffing system instances to provide a more complete view of the transit schedules and human mobility patterns. On top of the abstraction and aggregation modules, we construct an origin-destination matrix and pedestrian flow heat map to analyze transit efficiency in spatial and temporal domains.

2.3 System Implementation

The Wi-Fi monitoring system is operated on the Ubuntu 14.04.1 64 bit distribution (with Linux kernel version 3.19.0-28-generic), that runs on PC Engines APU platform [2]. The APU platform is a mobile embedded platform that is equipped with a 1GHz dual core CPU and 4G DDR3 DRAM. We conducted the sniffing tasks by using a multi-thread program written in C/C++. One thread runs the monitoring module to collect Wi-Fi packets from the specified wireless interfaces. Another module checks the correctness of received packets by validating the Cyclic Redundancy Check (CRC). The GPS module senses location changes and sends the GPS location information to a third thread. Both packets and GPS data are stored in SQLite database files. To protect public privacy, the private information included in each packet, e.g. MAC addresses, are hashed before saving into databases. And the real data is dropped immediately. The data analysis modules are written in Java.

2.4 System Deployment

We deploy our Wi-Fi monitoring system in two city buses. Those two buses have been assigned to three bus routes that are illustrated in Figure 3. The bus routes cover a large public university's main campus area as well as a residential area that accommodates graduate students and visiting scholars. The details of each route are shown in Table 1. The two city buses are operated by one local bus company.

These two buses are usually scheduled to be on the road from 6am to 6pm on route 80. Buses are also occasionally scheduled to

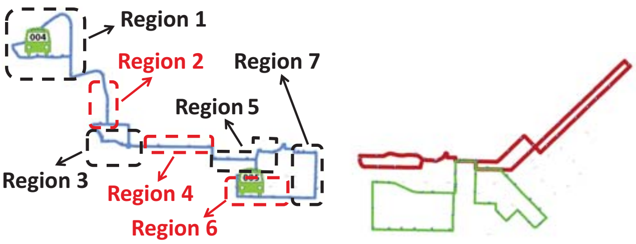


Figure 3: Bus routes with labeled bus stops. Route 80 (blue) map on the left, Route 81 (red) & 82 (green) on the right. The map size is roughly 1.5 mile × 2 mile. We segment the route for bus line 80 into seven disjoint regions for easy analysis.

Table 1: Route Statistics

	Route 80	Route 81	Route 82
Trip Distance (miles)	7.91	5.65	5
Trip Time (mins)	45[1]	30	30
Service Span	6am-3am	6:30pm-3am	6pm-3am
Total Station Number	47	31	34
Frequency (mins)	7-50[2]	30	30

[1] The actual trip time ranges from 40 to 50 minutes during different hours of a day.
[2] The frequency will change during different hours of a day, e.g. during rush hours, it will have a higher frequency to satisfy high volume riding demands.

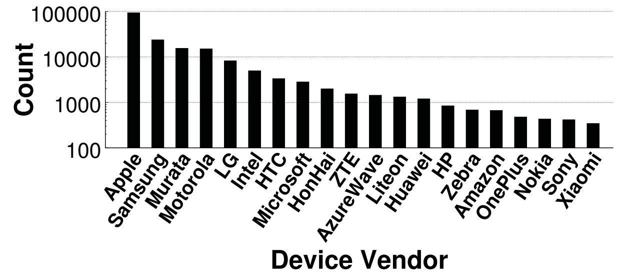


Figure 4: Distribution of devices by vendors in log scale.

Table 2: Collected Data Statistics

	Route 80	Route 81	Route 82
Days	258	23	24
Hours	3,225	126	65
Distance Covered (miles)	31,510	1,425	510

operate during night hours on route 81 and 82. Detailed statistical information about recorded data is summarized in Table 2. We collected data from both buses for around 300 days for 12 hours per day. In total, during these 300 days, two buses travel more than 32,000 miles. Among the collected data traces, the two buses ran on route 80, 81 and 82 for 258, 23, and 24 days accordingly. More than 300,000 unique Wi-Fi devices were detected by our system. By looking at the Organizationally Unique Identifier (OUI) [5] of the MAC address (the first three octets), we are able to compare the distribution of various vendors. As shown in Figure 4, Apple dominates all other vendors.

Starting from iPhone 5s and iOS 8, Apple introduces randomized MAC address in probe requests under certain settings to protect user privacy. According to Zebra Technologies' white paper [46], the MAC randomization can only be triggered when both cellular data and location service are off, with Wi-Fi turned on but not connected. We also performed the same kind of experiments using iPhone 6 with iOS 8 and Wireshark toolkit. And we have similar observations. According to recent studies [17, 32, 43], both iOS and Android devices can be re-identified and tracked. MAC randomization certainly overestimates the number of users, but it exposes limited impacts on statistical transit analytics.

3 OUR APPROACH TO TRACK INDIVIDUAL

In this section, we describe how to reconstruct bus schedules and passenger riding patterns.

3.1 Passenger and Pedestrian Tracking

Trellis keeps track of each individual by tracking their Wi-Fi enabled devices, and it separates different devices based on the MAC addresses included in the Wi-Fi packets. Figure 5 illustrates the architecture of Trellis. Trellis first determines the type of the device based on received 802.11 packet type, that is, Trellis identifies that the device is a Wi-Fi access point or a mobile device. If the device is a mobile device, further analysis will be conducted. As discussed in previous sections, when the vehicle is in motion, the signal strength of a passenger's Wi-Fi device as observed by a vehicle-mounted Wi-Fi observer should be fairly stable; while the signal strength of an outside pedestrian will vary in a predictable way before eventually disappearing. We developed two schemes to discern which device is inside the bus and which is not. A feature driven scheme is a straight forward identification mechanism. We set different thresholds on RSSI, distance and duration to determine who is inside the bus. However, it is hard to select one set of thresholds that can work under different scenarios. Hence, we extracted features from GPS and RSSI data, then used a hierarchical clustering algorithm to distinguish passenger and pedestrian.

Some unpredictable factors certainly affect our system accuracy. For example, some people may still use feature phones or have more than one smartphone; some people will turn off their Wi-Fi function to save power. Under these circumstances, our system will either overestimate or underestimate the total device number. However, we are focusing on the statistical trends of human activities, not the exact number of passengers and pedestrians. Our observation shows that we can still quite effectively track general trends in

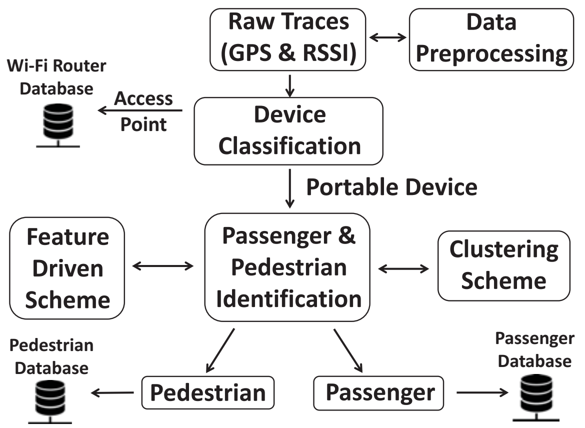


Figure 5: Trellis architecture. The raw GPS and RSSI is processed together to identify passenger and pedestrian. Passenger and pedestrian are sent to corresponding modules for further information extraction.

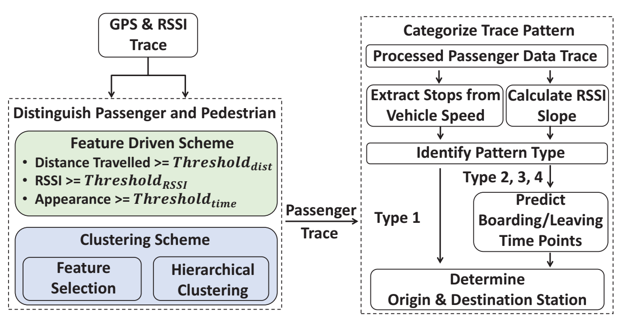


Figure 6: Illustration of two schemes and how to keep track of each passenger.

transit behavior from a long-term view. Next, we explain how Trellis gathers enough information for transit usage analysis.

3.1.1 Passenger Detection. The most challenging task for passenger detection is to extract useful information from collected data. First, the RSSI readings are highly fluctuating. Therefore, we cannot use RSSI alone as the indicator to identify if one passenger is on the bus. Second, the Wi-Fi signals are opportunistically received. The Wi-Fi signals' transmitting frequencies are based on user activities, such as screen being on and off. We have developed two schemes, a feature driven scheme and a clustering scheme to identify whether one subject is on the bus or not. Figure 6 provides an overview of the two schemes.

Feature Driven Scheme: We use multiple RSSI readings observed at different locations to determine the location of that subject. If there are consistent high RSSI readings from a specific device after the bus has been traveling a certain distance (or readings that appear for a certain period), this device is on the bus with high probability. We will discuss how to find the RSSI threshold δ_{on}, distance threshold β_{on} and duration threshold θ_{on} in section 3.1.4.

Clustering Scheme: As shown in Figure 10, emission power of on-bus devices varies greatly. Since the feature driven scheme uses a threshold-based algorithm, the classification results may be affected by some bias factors. To resolve this potential problem, we use a clustering algorithm to classify passenger and pedestrian. Here is a list of features we used for a hierarchical clustering algorithm.

Packet: Mean, Median, Standard deviation, Percentage of RSSI readings greater than -70, Packet receiving rate, $\frac{\sum RSSI}{Duration}$

GPS: $\frac{Total\ Distance}{Duration}$, $\frac{Total\ Distance}{Packet\ Size}$, Average speed, Speed standard deviation

To eliminate potential bias, all features are normalized to values between 0 and 1.

3.1.2 Passenger Tracking. We divided the entire bus route into continuous road segments, and each road segment is between two consecutive adjacent bus stations. Hence, each passenger travels with a bus for at least one road segment. Ideally, we can observe an increasing trend of RSSI values which would indicate that the passenger gets on the bus. Then RSSI readings stay stable for a period of time and then decrease when the passenger gets off the bus. Based on observation, RSSI & speed patterns can be categorized into four types. Figure 8 demonstrates four possible types of inference from RSSI & speed patterns.

Type 1: An ideal RSSI & speed pattern, the place and time that a passenger gets on and off the bus can be clearly identified. (Figure 8a)

Type 2: Only boarding information can be inferred. (Figure 8b)

Type 3: Only leaving information can be inferred. (Figure 8c)

Type 4: Neither boarding nor leaving information can be inferred. (Figure 8d)

Boarding and leaving information are hard to infer for patterns Type 2, 3, and 4. A prediction method has been developed to handle imperfect cases. (Figure 8d)

Determine Pattern Type: We determine the type of RSSI & Speed pattern by checking RSSI slopes with vehicle stop points (where vehicle speed is zero). First, we calculate the slopes of RSSI values using equation 1.

$$Slope_i = \frac{\triangle RSSI}{\triangle Time} = \frac{RSSI_{i+1} - RSSI_i}{t_{i+1} - t_i} \tag{1}$$

Ideally, the slope values should be positive at the beginning indicating the passenger gets on the bus and the slope values should be negative at the end. RSSI slope values are close to zero while the passenger is on the bus. Then, we extract vehicle stop points based on vehicle speed. Combining vehicle stop and RSSI slope information, passenger boarding and leaving points can be inferred. Figure 7 shows an example of RSSI slopes with vehicle stop points of Type 1 pattern. Slope values are positive in the beginning and negative in the end. The vehicle stops when the peaks appear, indicating the stations where the passengers get on and off the bus. Couple RSSI patterns with vehicle speed of movement, passenger riding information can be clearly inferred for this example. For Type 2 and 3 patterns, only positive or negative RSSI slope values could be observed in the beginning period or at the end of the trip correspondingly. All RSSI slope values are close to zero for Type 4 pattern.

Handling Type 1: For this scenario, the starting bus stop of the

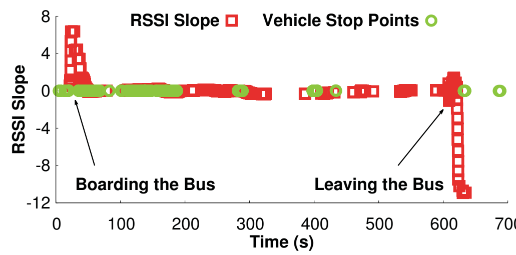


Figure 7: An example of how to determine Type 1 pattern from RSSI slopes and vehicle stop information.

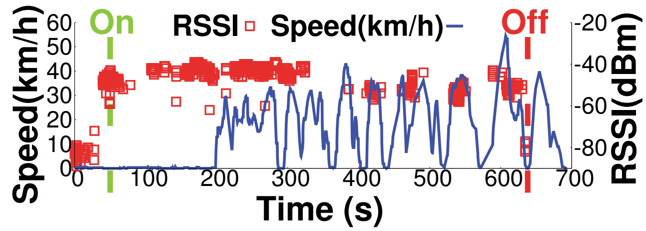
first road segment is recognized as the location where the passenger gets on the bus. The ending bus stop of the last road segment is recognized as the location where the passenger gets off the bus.

Handling Type 2, 3 and 4: When the phone screen is off, the system will reduce the frequency of sending out probe request packets to save power. Figure 10 illustrates the transmission rate for different brands of devices. Theoretically, the device should send out a probe request every τ seconds, which means we could detect that device at least once every τ seconds. It is possible that the passenger could get on and off the bus during this τ second period. This is why Trellis fails to identify the boarding or leaving or both locations for Type 2, 3, and 4 scenarios. To handle these scenarios, we developed a model to predict when the passenger gets on or off the bus. First, based on frequencies of the received packets from that specific device, we estimate τ using equation 2.

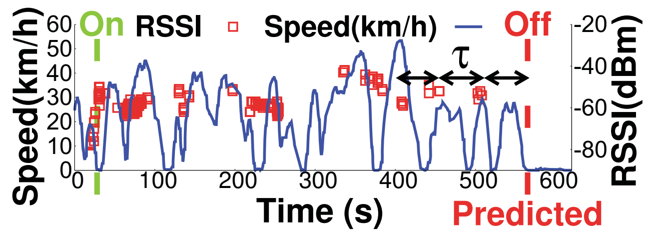
$$\tau = \Sigma_i^n f_i * Duration_i \tag{2}$$

where $Duration_i$ is the time difference between i-th and $(i+1)$-th packets, and f_i is the appearance frequency of $Duration_i$. Note that we eliminate the $Duration_i$ when it is less than 10 seconds. To determine boarding time, we begin by using the first half of the received packets to derive τ, and vice versa. Second, we explore all the places that the bus traveled during τ seconds (looking ahead for boarding station predictions, and looking behind for leaving station predictions). The system chooses the bus stop where the bus stops before/after τ seconds as the boarding/leaving bus stop. Figure 8b, 8c, and 8d show the estimated τ and predicted boarding and leaving time.

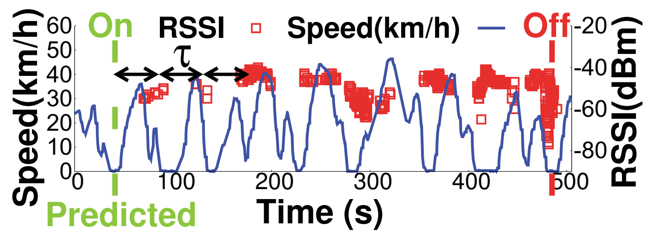
3.1.3 Pedestrian Detection. Pedestrians can be detected anywhere along the bus route. Our pedestrian identification algorithm first checks the total distance a bus traveled. If the travel distance is less than a distance threshold β_{off}, then it will check the RSSI readings. In contrast to the passenger RSSI readings, a portion of α readings should be less than the threshold δ_{on}. If an individual satisfies the above two conditions, it has a high probability of being on the road. As shown in Figure 10 (left), the RSSI readings from a device in a car tailgating the bus are much less than those from a passenger's device. Therefore, this device cannot be classified as a passenger device. What's more, the total distance a given bus traveled should be much longer than β_{off}. Thus, this device will be treated as a pedestrian device.



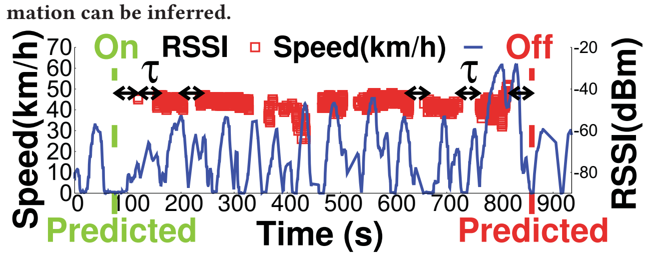
(a) An example data trace showing where a passenger gets on and off the bus.



(b) An example data trace showing only a passenger's boarding information can be inferred.



(c) An example data trace showing only a passenger's leaving information can be inferred.



(d) An example data trace showing that no passenger detailed information can be inferred.

Figure 8: Four possible types of inference from RSSI and speed data patterns for detected passenger.

3.1.4 Parameter Selection. Figure 9 summarizes the cumulative distribution functions (CDF) of station to station travel time and the distance of the three routes. Detailed information of each route is acquired from the GTFS [3] data published by the local metro company. From the figure we can see that 80% of the travel time between stations is more than 50 seconds. More than 80% of travel distance between stations is longer than 150 meters. For a feature driven scheme, we set the distance threshold β_{on} as 200 meters, choose 100 meters for β_{off}, and assign 1 minute to θ_{on}.

After we recognize one detected device is a passenger device, we can study the Wi-Fi module emission power and compute packet transmission rates for different devices of various vendors. The CDF

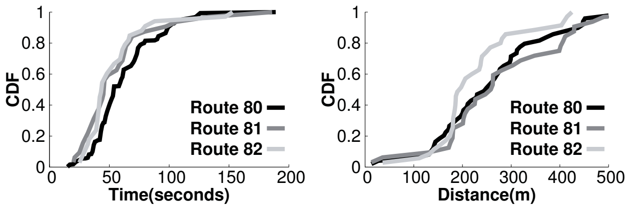


Figure 9: The CDF of station to station travel time (left) and distance (right).

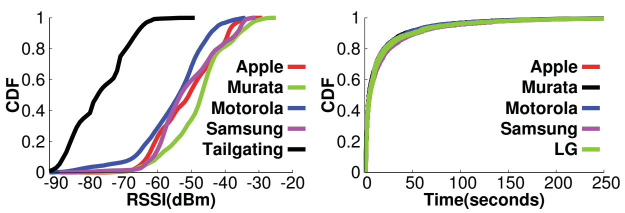


Figure 10: The CDF of mobile device Wi-Fi signals’ RSSI readings (left) and transmission rate (right).

of emission power of on-bus devices are summarized in Figure 10 (left). To plot this CDF, we use RSSI readings observed from on-bus devices. Due to limited space within this paper, we show only devices made by the four most popular vendors. In general, the signal strength for on-bus devices are greater than -65 for around 90% of time. In order to deal with the case of a car following the bus for multiple stops, we collected some data by driving a car tailgating the bus. We put an iPhone 6s Plus and a Nexus 5X in the car, and followed the bus at a close but safe distance. We did this experiment twice, each time for around a half an hour. The RSSI readings are shown in Figure 10 (left). The signal strengths from a device in the car following the bus are between -90 to -65. Hence, both the feature driven and the clustering scheme will not consider such a device to be a passenger device. What’s more, with the speed and GPS information, our system could tell this is not a pedestrian device (For normal people, it is very hard to run as fast as a bus for a long distance). Therefore, we test the feature driven scheme with δ_{on} ranging from -70 to -60 db. The CDF of the Wi-Fi signal transmission rates is shown in Figure 10 (right). Since the transmission rate is within 50 seconds for 90% of the time, this transmission resolution suggests that our individual tracking algorithm has the ability to handle corner cases such as when a passenger is on the bus for only one road segment.

3.2 Transit Schedule Reconstruction

In order to perform public transit analytics, e.g., route design, scheduling, etc., we track each bus on the route to record when it passes each stop. To reconstruct the transit schedule from collected data, we build stop tables for each route. Each bus stop in the table is labeled by an index, GPS location information and direction. All the information is gathered from the published GTFS feeds. Based on the vehicle driving direction as well as the location information, we can infer when the buses pass each station. This module essentially establishes when the bus arrives at each station and how long it stays there. This information is important for transit operators to evaluate the on-time performance of each bus.

3.3 Origin-Destination Matrix

For most kinds of analyses in the field of traffic planning and analysis, there is a need for origin-destination (OD) matrices, which specify the travel demands between the origin and destination nodes in the network. Hence, we built an origin-destination matrix, which essentially records how many passengers ride from one bus station to another. We divide the 47 bus stations in route 80 into seven geographically adjacent regions for easy analysis (as illustrated in Figure 3). Within each of the seven regions, there are 11, 4, 6, 7, 7, 5 and 7 bus stations, respectively. Based on this matrix, we can analyze passenger region-to-region movement patterns. Additionally, we may add other dimensions to understand passenger behaviors, i.e., time domain and weather conditions, to analyze passenger riding patterns during different periods of the day or under different weather conditions.

4 PASSENGER ACTIVITY TRENDS

In this section, we evaluate Trellis by demonstrating various transit usage analysis results.

4.1 Tracking Bus Occupancy

Bus occupancy is an important factor for transit operators to make transit plans, improve the transit efficiency and seek government funding. After reconstructing the transit schedules and identifying passengers, we gathered enough information to count passenger and record how many (essentially which) passengers get on and off at each bus station. Traditional methods such as questionnaires and bus driver counting could help transit operators understand riding patterns. However, these methods require a lot of human labor work, and are time consuming.

Our system provides a low-cost approach to assist or even replace existing counting methods. We evaluate the counting algorithms by comparing estimated passenger numbers and ground truth. To get ground truth data, we recruited several volunteers, and asked them to take the bus and count the number of passengers getting on/off the bus at each bus stop. Volunteers counted the numbers and recorded them using a customized Android application. We collect the ground truth data in 20 trips on 20 different days. For each trip, each volunteer stayed on the bus for around one hour. The aggregated ground truth data covers every hour from 9am to 9pm, including weekdays and weekends. The ground truth data was then synchronized with the data collected by the sniffing system based on time and GPS location.

Based on our observation, Trellis can, on average, detect 65% of the total passengers. Wang *et al.* reports that their system can discover around 40% of customers waiting in a queue [44]. Musa *et al.* claims that their system can detect a passing smartphone 69% of the time if the Wi-Fi is turned on [36]. Consider the increasing trend of smartphone penetration rates [40], we believe 65% is a fairly reasonable discovery rate. Hence, Trellis scales the estimated passenger number by 1.5. We summarize the calculated passenger

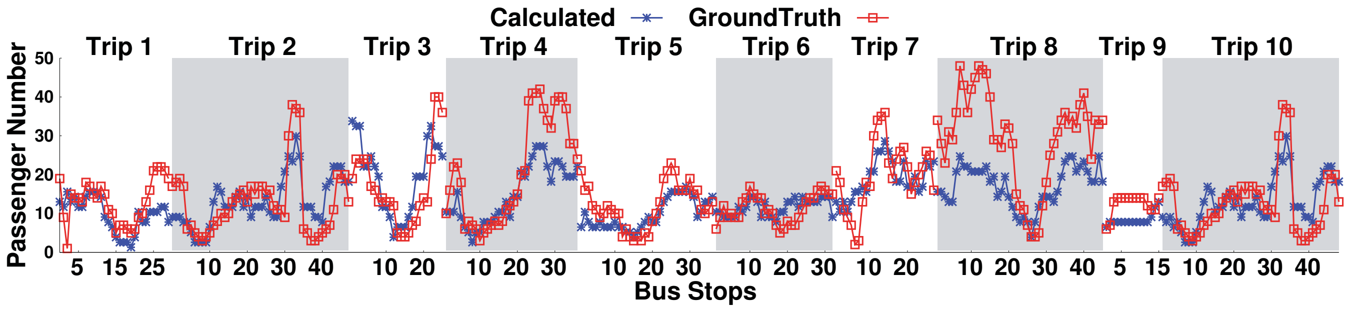


Figure 11: Onboard passenger number ground truth and automatic passenger counting results.

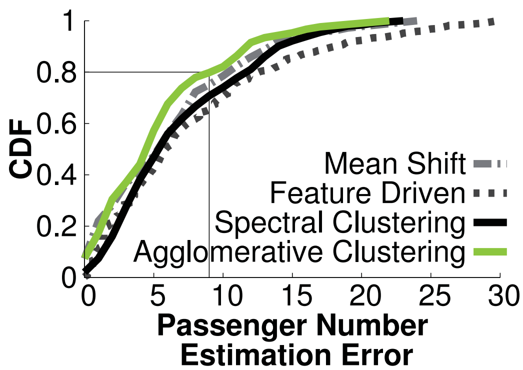


Figure 12: The CDFs of passenger number estimation error with different schemes.

numbers and the ground truth passenger numbers in Figure 11. The x axis represents the number of bus stops counted for that trip; the y axis shows the passenger number on the bus between two consecutive bus stops. Due to space limitations, we only show 10 trips in this figure. Each red point represents the actual number of passengers on the bus at that bus stop. The ground truth passenger numbers were counted at each station where the bus stopped, after existing old passengers left the bus and new passengers got on. The total passenger numbers stay the same between two consecutive stops. Each blue point refers to the total number of passengers calculated by Trellis. The estimated passenger number is the total number of passengers on the bus between two consecutive bus stops. Figure 11 shows that the major estimation error is caused by passenger bursts. The bursts usually occur when a large volume of students finishes one class together, gets on the bus at one stop, then gets off the bus at next stop together to take another class in a nearby building. Some students may not turn on phone screen during this short trip, so we may lose track of them. However, we still can see the rising and falling trends from estimated passenger numbers when bursts happened. Hence, the loss of tracking has little effects on long term statistical analysis.

Figure 12 summaries the passenger number estimation errors of different schemes. For the feature driven scheme, when the RSSI threshold δ_{on} is set to -65db, Trellis has the best performance. We tested with five clustering algorithms, namely the Affinity Propagation, Mean Shift, Spectral Clustering, DBSCAN and Agglomerative Clustering. Due to space limitations, we show the results of only the best three algorithms in Figure 12. Agglomerative Clustering has the best estimation performance. The passenger number estimation error is within 7 for around 80% of the time.

Key Observations: Regarding the evaluation of passenger counting algorithms, Trellis can discover 65% of total passengers. Passenger counting error is within 7 for 80% of the time.

4.2 Transit Riding Patterns

The ability to track each individual allows us to conduct transit statistical analysis. For instance, we can study passenger riding behaviors and discover various types of passenger riding patterns at different stops. As previously noted, the bus route covers a residential area and a main campus area, and our analysis shows that passenger riding habits are periodic during weekdays among bus stops in different regions. We summarize the average number of passengers getting on and off at two specific bus stops located in two regions during each hour of one week in Figure 13. The top graph shows the passenger riding patterns at a bus station located in the residential area (region 1 in Figure 3) and the bottom graph is for a bus station located in the main campus (region 5 in Figure 3). There are two main observations generated from these two figures. First, same riding patterns (including getting on and off) repeats from Monday to Friday, and changes during weekends. Second, bus stops located in different regions have different riding patterns. For instance, in the residential area, people are going out for work in the morning and going back home in late afternoon. Hence, there are obvious riding peaks during those hours. Further, undergraduate students live on campus. They travel between dormitories and campus buildings for different classes throughout the day, so there are peaks in the number of passengers getting on and off the bus throughout the day.

Key Observations: Regarding the study of passengers' riding behaviors at stops in a residential area as well as in the main campus area, at a residential bus station, most people get on the bus in the morning and get off the bus in the evening whereas in the main campus area, people get on and off the bus throughout the day.

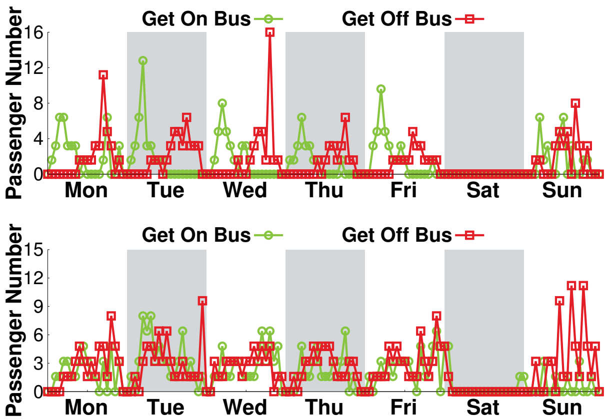


Figure 13: Riding patterns of different bus stops in the residential area (top) and the main campus (bottom).

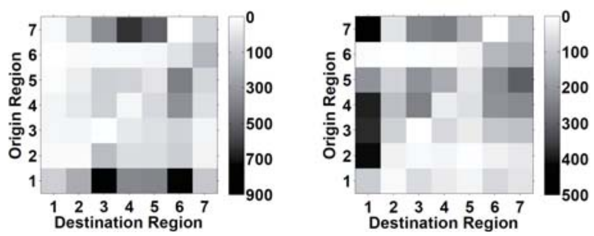


Figure 14: Original-Destination matrices during morning hours (left), and evening hours(right).

4.3 Transit Scheduling Analytics

We build OD-matrices using passenger region-to-region movements data. Figure 14 shows two OD-matrices during morning hours (7am to 9am) and evening hours (5pm to 7pm).

Suppose OD represents this OD-matrix and OD_{ij} denotes each element in the matrix (i represents the index of y axis and j represents the index of x axis). The value of OD_{ij} refers to the total number of passengers getting on the bus at bus stops located in region i and getting off at bus stops located at region j. Darkening colors correlate to increases in passengers. As can be seen from Figure 14 (left), most of the passengers travelling from region 1 are going to region 3 or region 6 during morning hours. From this observation, we can provide suggestions to operators. For example, extra direct buses connecting region 1 and 3 (or 6) can be added to the route during morning hours, which could reduce the travel time for passengers who want to go to region 3 or region 6 since the bus stops less frequently, while the rest of passengers can have a better riding experience due to less passengers being on the bus. During the evening rush hours, most of the passengers get on the bus from different regions and are going back to region 1, which means passengers are going back home.

Key Observations: By building OD-matrices for transit scheduling analysis and evaluation it is discernable that during morning hours, Regions 1 and 3 and Regions 1 and 6 are the most popular origin-destination pairs. Most traffic goes to Region 1 between 5pm and 7pm.

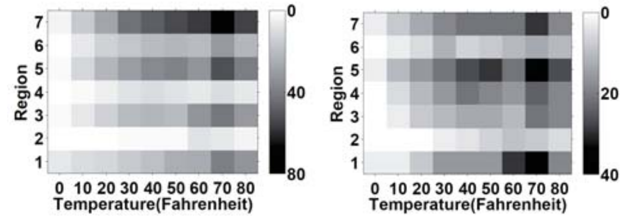


Figure 15: The impacts of temperature on daily average pedestrian (left) and passenger (right) number.

5 IMPACTS OF EXTERNAL FACTORS

In this section, we discuss how external factors, such as weather and temperature, impact human outdoor activities.

The weather data was collected using the Dark Sky Forecast API [8]. We requested hourly weather data and stored those data in a database. The requested weather data contains the following properties: icon (rain, clear, snow, etc.), precipitation intensity (inch/hr), precipitation probability, precipitation accumulation (inches), temperature (Fahrenheit), wind speed (mph), humidity and visibility (miles). Each hour's weather information entry in the database is indexed by a time integer key, e.g. 201512251300 represents Dec. 25, 2015 at 13:00.

5.1 Overview

Trellis was deployed in a northern city in the US. The data was collected through a mild summer and a severe winter. The temperature could be as high as 90° Fahrenheit in the summer but as low as −10° Fahrenheit during the winter. Therefore, outdoor temperature could be a key factor that affects human behaviors as well as traffic.

We built region-temperature matrices to show how would people react to different outdoor temperatures. We counted passenger and pedestrian numbers region by region, then, for each data entry, we queried the temperature at the time it was detected. Finally, we built the region-temperature matrices shown in Figure 15. The left one demonstrates pedestrian behaviors while the right one shows passenger behaviors. For the left figure, each box in the color map is the average daily number of pedestrians detected in that region at a certain temperature range, e.g. suppose the matrix is RT, then RT^1_{50} indicates the average daily pedestrian number in region 1 within the temperature range of 45° to 55° Fahrenheit. Similarly, for the right figure, each box in the color map is the average daily number of passengers that get on the bus from the station in that region at a certain temperature range. Darkening colors correlate to increases in the daily average number of people. As can be seen from these two figures, temperature affects both passengers and pedestrians in a similar way. For each region, there is an increasing trend of numbers as the temperature increases. When the temperature is 0° Fahrenheit (around −17° Celsius), we can rarely see people outside. As the temperature increases, the number of people increases, and reaches a maximum when temperature is around 70° Fahrenheit (around 21° Celsius).

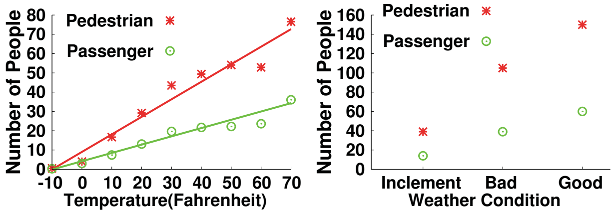


Figure 16: Quantify temperature and weather impacts on human activities in region 7.

Figure 17: The comparison of on-time performance between different hours(left) and weather conditions(right).

Key Observations: The results of this evaluation of temperature's impacts on human outdoor activities indicate a positive correlation between temperature and human outdoor activities.

5.2 Quantitative Analysis

To have a numerical intuition of the weather and temperature impacts, we conducted a quantitative analysis of these two factors based on the data we have. We compared the impacts of temperature on human mobility between rush hours and regular hours. Temperature has similar impacts on the human mobility during different time of the day (higher temperature, more human outdoor activities). One potential reason may be the temperature will not change too much during a day. As a result, we quantify the weather impacts using average daily number of people in this section. The figure on the left side of Figure 16 shows the relationship between the temperature and the number of people in region 7. We can see a positive correlation between the temperature and the number of people, i.e. as the temperature increases, more and more people are willing to go outside. To be more specific, when temperature increases ten degrees, around 15% more pedestrians show up on the street, and around 10% more passengers are present on the bus.

According to the Dark Sky Forecast API [8], we define weather conditions based on precipitation intensity. If it is raining or snowing and the precipitation intensity is greater than 0.35, we treat it as an inclement weather condition. If the intensity is between 0.05 and 0.2, then we regard it as bad weather. Finally, we identify good weather only when it is clear or sunny. We only tend to use these three conditions as a showcase for quantifying weather impacts. The figure on the right side of Figure 16 indicates there is a positive correlation between weather conditions and the number of people. More people participate in more outdoor activities as the weather improves.

Key Observations: Temperature and weather changes affect human outdoor activities.

5.3 Impacts on On-Time Performance

Traffic conditions vary dramatically during different times of the day and are affected by various factors. Under inclement weather conditions, especially when weather is snowy or icy, drivers increase headway, decrease acceleration rates, and reduce speeds, which collectively results in traffic congestion and schedule delay. Providing an efficient public transportation system can essentially alleviate traffic congestion.

In Figure 17, we summarize the on-time performance under different scenarios. For this comparison, we focus on the data collected from route 80. The figure on the left shows the difference between peak and off-peak hours. A negative value means that the bus arrived earlier than the scheduled time while a positive value represents how late the bus was. During rush hours, bus drivers need to reduce speeds and brake frequently due to the high volume of vehicles on the road. What's more, passenger demands also increase. Hence, schedule delay is likely to happen during rush hours. In the left figure, we can see the difference between two CDFs, which shows a longer delay during rush hours. Compared to regular hours, early or late arrivals happen more frequently during rush hours. During regular hours, the bus may arrive early or late within a 3-minute interval. For worst case scenarios during peak hours, passengers would experience an 8 minutes' wait, or the bus may arrive 7 minutes earlier than scheduled. One thing that needs to be mentioned is that the transit schedule already takes traffic conditions into consideration, i.e., transit operators schedule more buses on the road during rush hours. Hence, from the CDFs we can see the difference is not that significant, which reflects the effectiveness of current transit schedules. The right figure shows the CDFs of lateness under inclement, bad and good weather conditions. When bad weather happens, traffic slows down and drivers drive more carefully, which all may contribute to a schedule delay. It is shown that as the weather worsens, the delay increases. Under inclement weather conditions, the bus could have a 7 minutes delay for a worst case scenario.

Key Observations: Passengers experience a longer delay during rush hours or when bad weather happens.

6 PEDESTRIAN ACTIVITY TRENDS

Pedestrian activities are of great importance for both the design and evaluation of public transit. Traditional transit evaluation approaches lack ways to gather pedestrian information. Our system brings the possibility of using human mobility patterns to evaluate transit systems.

6.1 Pedestrian Activity Analysis

Pedestrian flows have been shown to be in close relationship with traffic flows [37], i.e. pedestrian flow can reflect and affect traffic conditions. Detecting pedestrian activity could help transit operators have a better understanding of traffic conditions.

6.1.1 Overview. Trellis conducts pedestrian detection concurrently with passenger tracking. As opposed to passengers on the

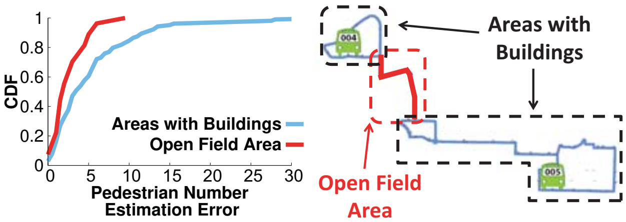


Figure 18: The CDF of pedestrian number estimation error.

bus, pedestrians are far away from the Wi-Fi monitor. Thus, the RSSI readings from a pedestrian device would be smaller than those of a passenger. Based on this fact, Trellis identifies pedestrians by checking the RSSI readings. Details of the detection algorithm are discussed in section 3.1.3. While this simple mechanism can miscount some people in a nearby building, this is a very limited negative effect. Based on our experiment results, the chance of such a person being detected is limited, since the distance between the bus and the building is long and the bus passes that building within seconds. To evaluate the accuracy of the pedestrian detection algorithm, we conducted two different kinds of experiments. First, we evaluated the detection accuracy by comparing the estimated pedestrian number with ground truth data. Second, we conducted experiments in an open field area, to study how people inside nearby buildings influence the estimation accuracy. Along route 80, there is an open field area where there are no buildings on both sides of the street. This area runs along a large lake, and it also contains a marsh and a softball field. This area is shown in Figure 18 (right); the bus route in this area is labeled in red. For the rest of route 80, buildings are located on both sides of the street. To get ground truth data, we asked volunteers to take the bu, and count the pedestrians on the street as the bus passed by. Figure 7 (left) shows the CDFs of the pedestrian number estimation error. In general, Trellis performs better in the open field area. Around 80% of times, the estimation error is within 7 people. The evaluation results show that Trellis performs well for most cases and will not be affected heavily by people inside a nearby building. Again, in this work, we focus on the general behavior of passengers and pedestrians, not exact numbers.

6.1.2 Time Impacts. Figure 19 explains the daily average pedestrian number along route 80 during the day and night. We used data collected from route 80 to plot these heat maps [2]. The top left part in each map represents the residential area while the bottom right part is the main campus. University hospitals and medical schools are located in the middle left. We can see major population centers appear differently throughout a day.

During the daytime, high population density areas are mainly distributed in the residential area, the hospital and some parts of the main campus. This is because various kinds of activities happen in those areas, such as going to school for classes and traveling between different campus areas. During the night hours, the library area is the only active area. Students gather in the library to do homework or participate in group studies. From the GTFS feeds,

[2]Route 81 and 82 only operate during night hours (6pm to 2am).

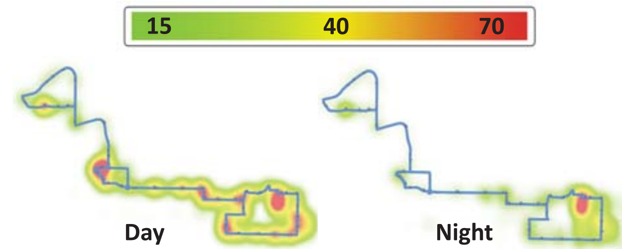


Figure 19: A comparison of the daily average pedestrian number on the street during daytime (10am-3pm) and night (9pm-11pm) hours.

service frequency varies from 7 minutes to 50 minutes during a day for route 80. During the morning and evening rush hours, the frequency is 7 minutes. And after rush hours, the frequency decreases from 7 to 14, and then to 25 minutes. Finally, the frequency shifts down to 50 minutes at night. Transit operators carefully design this schedule to serve public efficiently. Our pedestrian activity analysis observations follow and support these facts. We believe Trellis is reliable to perform these transit analytics effectively at a large scale.

Key Observations: By identifying pedestrian activity patterns during different times of day, we can locate several popular areas during the daytime and see that the library area is the only active area at night.

7 RELATED WORK

The concept of using a wireless-based approach for transit analytics was first considered in a recent position paper [28]. However, the prior work did not consider the broader opportunity of using this platform for analyzing pedestrians in city streets, nor did it consider the ability to analyze human mobility across different external factors, such as weather and changing temperatures, etc. More importantly, it did not conduct a significant evaluation of these opportunities in practical settings.

7.1 Passenger Counting

Transit operators need to collect transit usage statistics either by manually counting or using expensive sensor systems. They are required to submit usage information to national transit database [7]. APC presents a passive, non-radiating infra-red technology to detect and count people moving through a door or gate [19]. The system has the ability to detect the number of passengers, but it needs expensive hardware and is not able to track each individual that is riding between each pair of stations. Chen *et al.* [14] use a video based algorithm to count passenger numbers. Meanwhile, some Asian cities use a system that requires each passenger to tapping IC card when gets on and gets off the bus [4, 48]. These systems cannot count those passengers who are paying cash. More importantly, tapping card may cause extra delays and queues at each bus station. Trellis does not require any passenger operations.

7.2 Human Mobility Study

Many applications such as traffic engineering and urban planning need to understand human mobility [10]. Gonzalez *et al.* [20] demonstrated the regularity of human trajectories by tracking user smartphone. Zhang *et al.* [47] studied human mobility based on multiple data resources, e.g., cellphone and transit data, to avoid biased judgment by single data resources. [18] infers human mobility by using taxicab location traces. Our work did similar things and proposes new applications by performing individual monitoring. However, we propose a novel way to conduct public transit analytics by using Wi-Fi monitors on city buses, which separate our work from existing ones.

7.3 Human Tracking by Wi-Fi

Wang *et al.* proposed a system that can track human queue length based on received Wi-Fi signal features and analyze the waiting time in the queue [44]. Depatla *et al.* [16] estimate the total number of people in an area based on Wi-Fi device power measurements. However, this technique requires customers' smartphones to become connected with APs and to generate network traffic. VTrack [42] uses smartphone inertial sensors to estimate people's trajectory, which is fundamentally different from our approach. Musa *et al.* [36] deploys multiple monitors on the road to estimating the trajectory of smartphone holders.

7.4 Traffic Monitoring

Research work has been done on traffic monitoring using instrumented probe vehicles to get sparse probe data. Herring *et al.* [24] use an HMM based probabilistic modeling framework to estimate arterial travel time distributions using collected sparse probe data. Thiagarajan *et al.* [41] proposed a smartphone based system that can shorten the expected waiting times. The study uses smartphones' accelerometer sensors and GPS to determine user position, and estimate traffic conditions. VTrack [42] also uses smartphones as sensor platforms and presents an energy efficient algorithm to predict traffic delay. It proposes an HMM based map matching scheme and travel time estimation model. In contrast to Thiagarajan's work, the VTrack method uses Wi-Fi data instead of inertial data. Nericell [35] takes advantage of mobile phones' multiple sensor readings and proposes a system that can monitoring road and traffic conditions. It requires the smartphone to send various sensor data, which may not work on all phones, and it is energy inefficient. In our study, we use a low-cost passive Wi-Fi monitoring system to get sparse probe data, and we are focusing on the different factors that can help improve transit efficiency.

8 DISCUSSION

In this section, we discuss the limitations of our system and propose other potential applications.

8.1 Limitations

These limitations expose challenges for our tasks, but they do not affect the practicability of our system.

8.1.1 Tracking Accuracy. As has been previously noted, passenger tracking and pedestrian detection accuracy are limited by some unpredictable factors. For example, some passengers are still using feature phones or their Wi-Fi function is turned off. For these cases, the Trellis system is not able to detect the presence of the individual. Some passengers may have multiple Wi-Fi enabled devices, e.g., a tablet and a smartphone. In this case, Trellis may overestimate the number of passengers. Although we use distance and RSSI thresholds to filter passengers and pedestrians, people driving a car following or parallel to the bus may be detected and cause overestimation. Either overestimated or underestimated device numbers will affect the system accuracy. However, since our goal focuses on the statistical trends of transit systems, such impacts can be ignored from a long-term view.

8.1.2 MAC Randomization. Apple introduced MAC randomization in iOS 8. Full implementation of MAC randomization is available in Android version 6.0 and above [1]. The key idea of this technique is to use fake MAC addresses in their probe request packets. It is hard for our system to identify this fake MAC address. However, from our own observations, the randomization feature in iOS requires several prerequisites to be triggered. Various discussions and reports [34, 46] also claim the similar findings. According to recent studies [17, 43], iOS devices can be re-identified by checking sequence numbers and timing information contained in the probe request packets. According to US Naval Academy's recent study [32], most Android phones simply do not have this technology enabled, despite the fact that they are running new versions of the operating system that should allow for it. It is even possible to track 100% of all test smartphones, despite the devices using randomized MAC addresses. Again, our work focuses on providing statistical analysis on transit efficiency to assist public transit operators instead of tracking every single passenger.

8.1.3 Pedestrian Detection. Our system can detect pedestrians along the route, but it has no capability of tracking each detected pedestrian. However, if we could deploy Trellis on more buses in the city metro network, we might have the ability to track pedestrians by building a communication network among buses. Therefore, several buses can track a single pedestrian and rebuild the pedestrian movement pattern. Our pedestrian detection algorithm may miscount some people in a nearby building along the routes. According to our evaluation results, such counting errors have little affect from a long-term view.

8.1.4 Smartphone Penetration. Another issue to note is the penetration of smartphones with built-in Wi-Fi through the passenger community. While it is hard for us to measure the true distribution of Wi-Fi devices within the passenger community, we can expect that older people and babies are less likely to carry smartphones. However, smartphones with built-in Wi-Fi are increasingly popular and accessible to larger populations. From a long-term view, the riding demand and trend of the passenger community will hold.

8.2 Tap-based Fare System

Tap-based fare systems are available all over the world [45]. Transit operators employ two strategies to collect riders' payments. One way is to set a fixed price for the whole route; the other way is to set the price based on the distance traveled. For the first mechanism, passengers only need to "tap in" to the transit system at their origin station. A distance-based pricing strategy requires passengers "tap in" at their origin station and then "tap out" at their destination station. Tap-based fare systems are more efficient than traditional conductors but cause customer delay during peak hours [9]. More recently, Metro Transit riders are able to pay transit fares using their smartphones [39]. Mobile applications have been developed for passengers to buy tickets online. Trellis could be part of such mobile applications. When a passenger enters the station, Trellis could determine his/her origin and destination, then calculate the price for each trip the passenger has taken. In the meantime, Trellis could evaluate transit usage and send real time statistics to transit operators. Since the whole process can be done automatically by the mobile application, passengers do not need to physically "tap in" and "tap out", thus minimizing queuing delays.

8.3 Other Applications

Currently, Trellis is deployed on two city buses, and it has been assigned to three routes. As we continue to cooperate with the local metro transit, Trellis will run on more buses and routes simultaneously, which will further extend its applications. For instance, with data collected from several buses on different routes, we could provide evaluation on transit interchange performance.

We mainly focus on providing transit analytics in this work, however, many more applications could be developed given this rich data set. For example, it is possible to predict the riding route of each individual passenger. Wi-Fi related applications can also benefit from an accurate predication of a passenger's presence [31].

9 CONCLUSION

This paper presents Trellis, a low-cost, vehicle-mounted wireless monitoring system that can track passenger movements, detect pedestrian flows, and evaluate how external factors impact human mobility, thereby providing useful analytics to transit operators, and potentially urban planners. Trellis uses a passive approach to gathering individual mobility data. The system uses RSSI readings coupled with vehicle location information to distinguish passengers from pedestrians, and it can monitor each detected individual. This capability gives us the flexibility for evaluating a public transportation system across a city in a very cost effective manner.

In this paper, we attempted to provide analytics of specific aspects of passengers on transit vehicles and of human mobility as observed from these vehicles. With ParaDrop, Trellis can be easily deployed and managed in vehicles at a large scale. Various transit analytics can be derived on-board quickly. We believe this approach of observing human populations at city scales has many more interesting and useful applications. Such explorations will form a part of our future work.

10 ACKNOWLEDGMENTS

We would like to acknowledge the volunteers who helped us in collecting transit usage information. We are also grateful to the anonymous reviewers whose feedback helped bring the paper to its final form. This research project is supported in part by the US National Science Foundation through awards CNS-1405667, CNS-1345293, CHE-1230751, CNS-1343363, CNS-1555426 and CNS-1525586.

REFERENCES

[1] 2015. Android 6.0 Changes. https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html. (2015).
[2] 2017. APU Platform. (2017). http://www.pcengines.ch/apu.htm
[3] 2017. General Transit Feed Specification (GTFS). https://developers.google.com/transit/gtfs/. (2017).
[4] 2017. IC Card in Hong Kong Transit. http://www.octopus.com.hk/home/en/. (2017).
[5] 2017. IEEE OUI Registry. http://standards-oui.ieee.org/oui.txt. (2017).
[6] 2017. InfoDev Automatic Passenger Counting. http://www.infodev.ca/vehicles/counting-passengers.html. (2017).
[7] 2017. National Transit Database. *http://www.ntdprogram.gov/* (2017).
[8] 2017. The Dark Sky Forecast API. https://developer.forecast.io. (2017).
[9] Justin Antos and Michael D Eichler. 2016. Tapping into Delay: Assessing Rail Transit Passenger Delay with Data from a Tap-In, Tap-Out Fare System. *Transportation Research Record: Journal of the Transportation Research Board* 2540 (2016), 76–83.
[10] Fereshteh Asgari, Vincent Gauthier, and Monique Becker. 2013. A survey on Human Mobility and its applications. *arXiv preprint arXiv:1307.0814* (2013).
[11] Paramvir Bahl and Venkata N Padmanabhan. 2000. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Vol. 2. Ieee, 775–784.
[12] Avishai Ceder. 2016. *Public transit planning and operation: Modeling, practice and behavior*. CRC press.
[13] Chao-Ho Chen, Yin-Chan Chang, Tsong-Yi Chen, and Da-Jinn Wang. 2008. People counting system for getting in/out of a bus based on video processing. In *Intelligent Systems Design and Applications, 2008. ISDA'08. Eighth International Conference on*. IEEE.
[14] Chao-Ho Chen, Tsong-Yi Chen, Da-Jinn Wang, and Tsang-Jie Chen. 2012. A cost-effective people-counter for a crowd of moving people based on two-stage segmentation. *Journal of Information Hiding and Multimedia Signal Processing* (2012).
[15] Yu-chung Cheng, Mikhail Afanasyev, Patrick Verkaik, Péter Benkö, Jennifer Chiang, Alex C Snoeren, Stefan Savage, and Geoffrey M Voelker. 2007. Automating cross-layer diagnosis of enterprise wireless networks. In *In Proc. of ACM SIGCOMM*.
[16] S. Depatla, A. Muralidharan, and Y. Mostofi. 2015. Occupancy Estimation Using Only WiFi Power Measurements. *IEEE Journal on Selected Areas in Communications* (July 2015).
[17] Julien Freudiger. 2015. How talkative is your mobile device?: an experimental study of Wi-Fi probe requests. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 8.
[18] Raghu Ganti, Mudhakar Srivatsa, Anand Ranganathan, and Jiawei Han. 2013. Inferring human mobility patterns from taxicab location traces. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM.
[19] Horst E Gerland and Kurt Sutter. 1999. Automatic passenger counting (apc): Infra-red motion analyzer for accurate counts in stations and rail, light-rail and bus operations. *INIT GmbH Innovations in Transportation* (1999).
[20] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. 2008. Understanding individual human mobility patterns. *Nature* (2008).
[21] S. He and S. H. G. Chan. 2014. Sectjunction: Wi-Fi indoor localization based on junction of signal sectors. In *Communications (ICC), 2014 IEEE International Conference on*. 2605–2610.
[22] Suining He, S-H Gary Chan, Lei Yu, and Ning Liu. 2015. Fusing noisy fingerprints with distance bounds for indoor localization. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 2506–2514.
[23] Dirk Helbing, Rui Jiang, and Martin Treiber. 2005. Analytical investigation of oscillations in intersecting flows of pedestrian and vehicle traffic. *Physical Review E* 72, 4 (2005), 046130.
[24] Ryan Herring, Aude Hofleitner, Pieter Abbeel, and Alexandre Bayen. 2010. Estimating arterial traffic conditions using sparse probe data. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE, 929–936.
[25] Rui Jiang, Qingsong Wu, and Xiaobai Li. 2002. Capacity drop due to the traverse of pedestrians. *Physical Review E* 65, 3 (2002), 036120.
[26] Rui Jiang and Qing-Song Wu. 2006. Interaction between vehicle and pedestrians in a narrow channel. *Physica A: Statistical Mechanics and its Applications* 368, 1 (2006), 239–246.
[27] Sheng Jin, Xiaobo Qu, Cheng Xu, and Dian-Hai Wang. 2013. Dynamic characteristics of traffic flow with consideration of pedestriansâĂŹ road-crossing behavior. *Physica A: Statistical Mechanics and its Applications* 392, 18 (2013), 3881–3890.
[28] Lei Kang, Bozhao Qi, and Suman Banerjee. 2016. A Wireless-Based Approach for Transit Analytics. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*. ACM, 75–80.

[29] Thomas Kimpel, James Strathman, David Griffin, Steve Callas, and Richard Gerhart. 2003. Automatic passenger counter evaluation: Implications for national transit database reporting. *Transportation Research Record: Journal of the Transportation Research Board* (2003).

[30] Peng Liu, Dale Willis, and Suman Banerjee. 2016. ParaDrop: Enabling Lightweight Multi-tenancy at the Network's Extreme Edge. In *Edge Computing (SEC), IEEE/ACM Symposium on*. IEEE, 1–13.

[31] Justin Manweiler, Naveen Santhapuri, Romit Roy Choudhury, and Srihari Nelakuditi. 2013. Predicting length of stay at wifi hotspots. In *INFOCOM, 2013 Proceedings IEEE*. IEEE.

[32] Jeremy Martin, Travis Mayberry, Collin Donahue, Lucas Foppe, Lamont Brown, Chadwick Riggins, Erik C Rye, and Dane Brown. 2017. A Study of MAC Address Randomization in Mobile Devices and When it Fails. *arXiv preprint arXiv:1703.02874* (2017).

[33] Virginia Miller. 2016. Americans Took 10.6 Billion Trips on Public Transportation in 2015. http://www.apta.com/mediacenter/pressreleases/2016/Pages/160331_Ridership.aspx. (2016).

[34] Bhupinder Misra. 2014. iOS8 MAC Randomization - Analyzed! http://blog.mojonetworks.com/ios8-mac-randomization-analyzed/. (2014).

[35] Prashanth Mohan, Venkata N Padmanabhan, and Ramachandran Ramjee. 2008. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM, 323–336.

[36] ABM Musa and Jakob Eriksson. 2012. Tracking unmodified smartphones using wi-fi monitors. In *Proceedings of the 10th ACM conference on embedded network sensor systems*. ACM.

[37] Takashi Nagatani. 2002. The physics of traffic jams. *Reports on progress in physics* 65, 9 (2002), 1331.

[38] Jacob Poushter. 2016. Smartphone ownership and Internet usage continues to climb in emerging economies. *Pew Research Center* (2016).

[39] Jake Sion, Transit App, Candace Brakewood, and Omar Alvarado. 2016. Planning for New Fare Payment Systems: An Equity Analysis of Smartphone, Credit Card, and Potential Mobile Ticketing Adoption by Bus Riders in Nassau County. In *Transportation Research Board 95th Annual Meeting*.

[40] Aaron Smith. 2017. Record shares of Americans now own smartphones, have home broadband. *Pew Research Center* (2017).

[41] Arvind Thiagarajan, James Biagioni, Tomas Gerlich, and Jakob Eriksson. 2010. Cooperative transit tracking using smart-phones. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 85–98.

[42] Arvind Thiagarajan, Lenin Ravindranath, Katrina LaCurts, Samuel Madden, Hari Balakrishnan, Sivan Toledo, and Jakob Eriksson. 2009. VTrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *Sensys*. ACM.

[43] Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo S Cardoso, and Frank Piessens. 2016. Why MAC address randomization is not enough: An analysis of Wi-Fi network discovery mechanisms. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. ACM, 413–424.

[44] Yan Wang, Jie Yang, Yingying Chen, Hongbo Liu, Marco Gruteser, and Richard P Martin. 2014. Tracking human queues using single-point signal monitoring. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. ACM.

[45] Wikipedia. 2017. List of smart cards – Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/List_of_smart_cards. (2017). [Online; accessed 2017].

[46] Zebra Technologies. 2015. Analysis of iOS 8 MAC Randomization on Locationing. *http://mpact.zebra.com/documents/iOS8-White-Paper.pdf* (2015).

[47] Desheng Zhang, Jun Huang, Ye Li, Fan Zhang, Chengzhong Xu, and Tian He. 2014. Exploring human mobility with multi-source data at extremely large metropolitan scales. In *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM.

[48] Pengfei Zhou, Yuanqing Zheng, and Mo Li. 2012. How long to wait?: predicting bus arrival time with mobile phone based participatory sensing. In *Mobisys*. ACM.