

# AwesomeBus: Increasing Awareness of Public Transportation Options

Eric Zeng, Lucy Simko

## ABSTRACT

Seattle has an extensive public transportation system, which will continue to grow in the coming decades. However, successful physical expansion of the transit network also requires public awareness and interest in the growing network. Currently, there exist popular tools to generate directions for point-to-point trips and to see real-time arrival data at any given stop, but mass transit riders lack a comprehensive tool for intuitively exploring the entire transit network without planning a specific trip or looking at real-time data for a specific stop. Our work, AwesomeBus, is a d3.js visualization that provides an interactive tool to explore Seattle's public transportation system. AwesomeBus enables exploration by showing every transit route and stop in King County on a detailed geographic map, and helps the user make sense of the information through intuitive visual encodings and multiple selection and filtering techniques.

## 1. INTRODUCTION

Mass transit ridership in the Seattle area has increased steadily in the past two years, driven by population growth, heavier automobile traffic, and expanded bus and light rail service [8]. As local transit agencies continue to add and revise bus and rail lines, transit riders need to be aware of the options that are available, and the destinations that are easily accessible via transit to increase utilization of the new systems.

This kind of exploration is crucial for potential or current riders who are considering moving to a new neighborhood, or who are new to Seattle entirely. For example, a graduate student who is moving to Seattle knows that she will be commuting daily to the University of Washington and may want to ensure she lives close enough to a transit route that runs directly to campus at all times of the day, so that she can commute efficiently around her (possibly irregular) lab schedule. After a year, this student may want to move neighborhoods once she discovers what she enjoys about Seattle's culture, but she still needs to stay close to a desirable transit line.

Another potential user is a tourist who is willing to walk somewhat more than someone who is trying to commute efficiently, but who wants to generally make his way to or from a certain neighborhood. Finally, we imagine that this kind of exploration will be useful for someone who typically does not use public transit to commute, but may be interested in understanding her alternate options, in case of road closure or mechanical issues with her car or bicycle.

Existing tools for visualizing public transit only provide

a limited picture of the transit system as a whole. Rider-focused visualizations include route timetables, route maps, static system-wide maps, real-time arrivals, and interactive point-to-point directions, which will be described in detail in Section 2. While each of these serves a particular purpose well, a common limitation is that none provide a high-level overview of all of the available routes and geographical areas serviced by transit, and they do not encourage intuitive exploration of new routes and stops.

Our goal in this project is to create a visualization of Seattle's public transportation system that displays specific transit information within the context of the rest of the transit system and the area-specific geography, encouraging rider exploration and discovery of new routes and destinations.

## 2. RELATED WORK

King County Metro and Sound Transit provide official timetables and route maps for individual routes [9] [11]. These are great for obtaining information about a specific route, like when a route is scheduled to arrive at a particular stop, and which stops the route services. However, the route maps are usually simplified geographical abstractions that omit context about adjacent streets and where connecting routes go. Transit riders who want to understand the transit system around their home would need to switch between multiple timetables and map representations to build a mental model of the local bus network.

One map that attempts to solve problem of providing transit network context is the the Seattle Transit Map [12]. It is a static, non-interactive map that shows all of Seattle's transit network on the map. Since there are over 200 routes to display on the map, it uses several techniques to help readers distinguish and filter between routes. Most notably, it uses line thickness to indicate the frequency of the route, and colors to indicate the mode of transit (i.e. light rail, streetcar, RapidRide)[13]. However, because the map is not interactive, it is still somewhat difficult to trace the path of a route, because routes often intersect and overlay with other routes of the same color.

OneBusAway shows a different type of transit data: real time arrival information. It shows when buses and trains are scheduled to arrive at stops in the near future, and how early or late each vehicle is expected to be. To view this information, users select a stop on a map, and then the app displays arrival times in the next hour. It is an extremely useful service for commuters and transit users who already know which stops and routes they will take, but people who are unfamiliar with the transit system need to use another

tool to discover routes and plan trips.

Another genre of transit visualization includes maps that use transit data as a way to expose other information about a city, such as the visualizations mentioned in [10]. These visualizations are in response to a contest held by the Urban Data Challenge; the winning visualization [1] shows transit usage for the user’s choice of bus line and day. The two runner-ups use transit data to highlight issues that are fascinating, but do not help the user understand anything about how she herself can use the transit network: one project contrasts poverty levels with the amount of transit service for each area of San Francisco [3]; the other runner-up displays bus stops where riders may be frustrated and uses color to distinguish why they may be frustrated (delay, capacity, speed) as well as size to show the level of frustration [4]. Another group developed interactive visualizations of Boston’s transit system, including a graph of trips by trains throughout the day, and a heat map showing times at which most people enter and exit stations [5]. All of these visualizations are informative and visually appealing, but they do not help a potential or uninformed transit rider understand the transit routes that are most useful to him.

### 3. METHODS

In this section, we discuss the implementation details and visualization techniques of AwesomeBus.

#### 3.1 Transit Data

King County Metro freely provides data on its routes and schedules using the General Transit Feed Specification (GTFS). GTFS data is divided into several relational data tables, such as stop locations, routes, trips, and shapes of routes. The data we are primarily interested in visualizing are the geographic path of the routes, the locations of stops, and the routes that serve each stop. To display this data in d3.js, we needed convert the relational GTFS data into GeoJSON LineStrings (sequences of line segments) to display routes, and GeoJSON Points to display stops [6].

For example, to encode a route in GeoJSON, we need to know the route number and all of the line segments that make up the route. However, the relationship between these two pieces of information are more complicated in GTFS. Since each named bus route can take multiple paths, depending on the day, time of day, and whether it continues as another route, this information is split between the routes, trips, and shapes table. *Routes* include high level information about the route, like its name, whereas *trips* represent individual trips taken by vehicles at different times of the day, and *shapes* contain the geographic coordinates traced out by vehicles.

Our data processing pipeline consists of four steps. First, we import the GTFS files into a database. Second, we run SQL queries to retrieve and join the relevant data from multiple tables. Third, we convert the data from its relational form into static JSON objects, which condenses the rows into mappings between an identifier (route name or stop name) and its geographic coordinates. Finally, we convert each of the intermediate JSON objects into GeoJSON objects, which can be rendered using d3.js. The processed GeoJSON data is saved to a static file and served directly to web clients, so that each page load doesn’t require the queries and processing code to be run again.

#### 3.2 Mapping

We render the transit data in the browser using a combination of d3.js and Mapbox. D3 retrieves the static GeoJSON data from our server, transforms geographic coordinates into screen coordinates, and renders them as elements on an SVG. However, d3 only renders the data that we provide, not additional contextual information like streets, water, and land masses. We decided to use Mapbox, an OpenStreetMap viewer, because it handle the base map layer as well as standard navigation features like zooming and panning. We chose Mapbox over other solutions like Google Maps, or using d3 to render contextual map information. Unlike Google Maps, MapBox allowed us to overlay the d3 SVG image on the map layer, allowing us to continue using d3 instead of rewriting our visualization code using another API. Mapbox provides a lot of features not included in stock d3, such as zooming and panning, and rendering tiled map data.

#### 3.3 Visualization Techniques

We apply standard visualization techniques to this public data set in a new way in order to provide a tool that reduces the gulfs of execution and evaluation for everyday transit users who are exploring transit options.

##### 3.3.1 Gulf of Evaluation

Existing visualizations make it difficult to understand the transit network as a whole. Official maps [9], directions [7], and real time data [2] only show a couple routes or stops at a time, which forces users to remember previous information as they explore transit options. Static system visualizations like the Seattle Transit Map [12] show all of the system information at once, but they contain so much information that it can be difficult to glean useful information. Our work addresses these evaluation issues by displaying a huge depth and breadth of transit route information and using standard data visualization techniques in order to not overwhelm the user.

**Color:** We primarily use color to distinguish routes. Initially, all routes are brightly colored, normal width, and almost fully opaque, in order to display all routes but also allow the overall geography of the area to be clearly visible. When a route is selected, it becomes fully opaque and slightly wider, and deselected routes are set to a neutral gray-blue with a decreased luminance and opacity in order to still give the necessary visual contextual information but ensure that the selected route can be preattentively processed by the user. For the Link Light Rail and the Rapid Ride transit routes, we use the established colors, dark blue and red, respectively. We also color the information in the side bar that corresponds to each route with the same color as the route on the map.

**Filtering:** In order not to overwhelm users, we take several steps to minimize the information drawn on the map. We only display bus stops at closer zoom levels, so that the map isn’t crowded with stops at low zoom levels. We also display a tooltip showing the route number(s) when users hover over a stop or selected route, rather than inscribing route numbers on the map. Lastly, we allow users to filter routes by time of day, and set a reasonable default, so that routes that only run in the dead of night are not shown initially. In addition to color, these techniques help the user visually distinguish between routes.

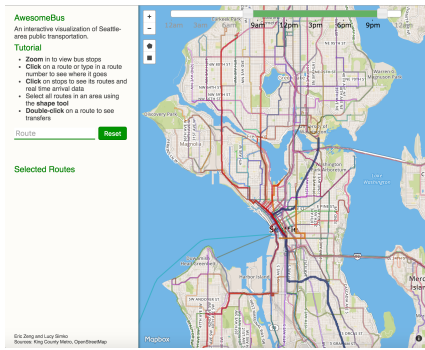


Figure 1: An overview showing all routes.

**Supplemental Text:** We reserve some of the left-hand side of the screen for a sidebar, which displays a short tutorial as well as a text box with which users can search for routes by name. Additionally, this sidebar displays links to real-time data on OneBusAway’s website when a bus stop is selected, and links to the official time tables for any selected routes. When routes are selected, the links displayed are the same color as the corresponding route on the screen. This provides the user with a wealth of potential additional information without displaying too much at one time. The colored links also act as a legend to help the user more quickly see which routes are which colors without mousing over every route.

### 3.3.2 Gulf of Execution

Existing transit visualizations do not encourage exploration of transit routes in part because they limit the user to asking a very specific question, like when bus X will arrive, or how to get between two addresses. Users may want to explore more complex types of information, but the existing tools don’t allow them to easily view multiple routes or select routes by geographical location. AwesomeBus addresses these issues by showing all route and stop information at once, and providing users with many different tools to intuitively select and filter routes and stops.

**Map Navigation:** Because users are used to interacting with geographical maps in certain ways, we use conventional navigational techniques: swiping or scrolling to zoom, and clicking and dragging to pan.

**Selection:** We also employ conventional selection techniques by allowing users to select routes by clicking on the route, typing the route name in the side bar, as well as drawing a selection box over multiple routes.

1. Visually selecting one or more routes
2. Visually selecting multiple routes in a geographic area
3. Querying for a route by its number
4. Selecting all of the routes that serve a single stop
5. Selecting all of the routes that can be transferred to from another route

We describe the specific implementation of these selection methods in section 4.

## 4. RESULTS

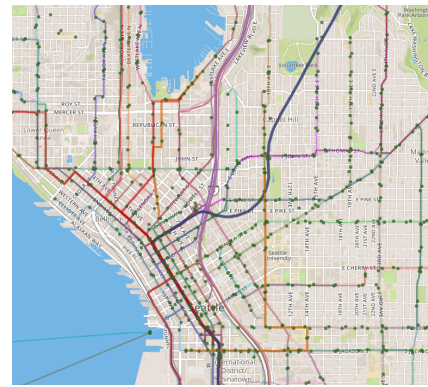


Figure 2: Zoomed in view showing bus stops.

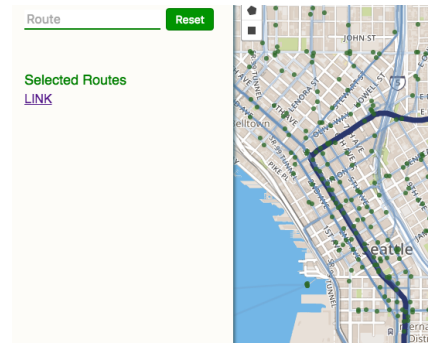


Figure 3: A single route is selected.

In this section, we describe the features built into AwesomeBus. Figure 1 shows the initial state of AwesomeBus, in which the map is zoomed to approximately the city limits of Seattle, and all routes are colored with thin lines that are not fully opaque. As the zoom level increases (see Figure 2), individual bus stops become visible.

Users can select routes by clicking on one of the lines (see Figure 3). This partially hides non-selected routes, and shows information about the selected route in the sidebar. Users can click and select multiple routes at once.

If the user clicks on a stop, all routes that serve that stop will be selected, and information about the stop will be displayed in the sidebar, like the stop’s name, and a link to real time arrival data from OneBusAway (see Figure 4). Note that the sidebar contains color-coded links to individual schedules for the routes that run through the selected

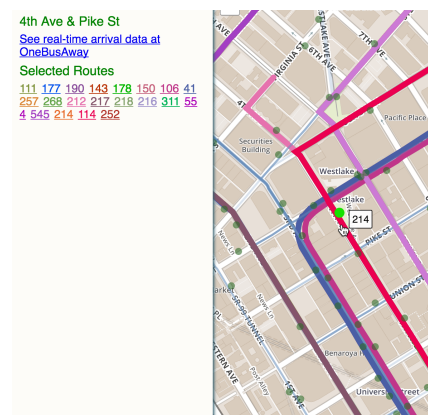


Figure 4: A bus stop is selected.

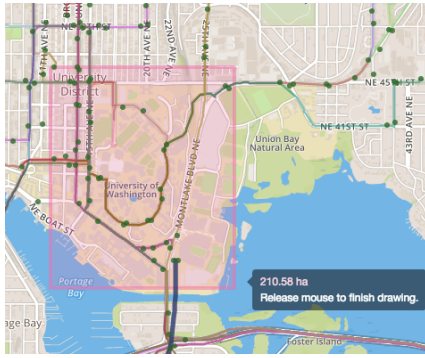


Figure 5: The user is selecting all buses that run through the area near the University of Washington’s campus.

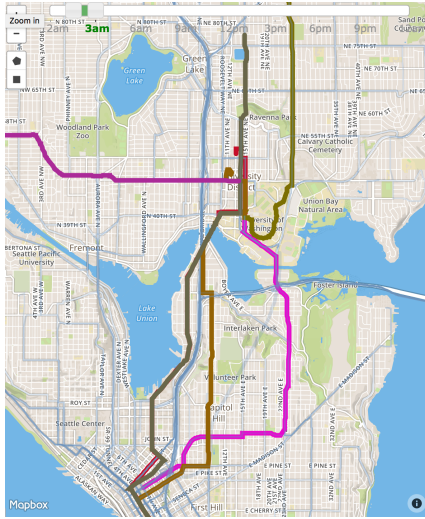


Figure 6: By moving the slider at the top of the screen, the user has revealed the routes that run between 2am and 4am in the area selected in figure 5.

stop.

If users know the name of the route but not the exact location, they can search for routes by typing the name into the search box on the sidebar, visible in figures 1 and 3.

To select multiple routes at a time in an area, as in figure 5, transit riders can use the shape drawing tool, displayed on the upper left side of the map near the zoom buttons, to select all bus routes that pass through a given area. This is useful as it allows a user to select an entire neighborhood to see how well it is connected to the rest of the city. This tool is also useful at a smaller scale, for a user to visualize the buses available to them if they walk  $n$  blocks from their house, or for a cyclist who is willing to ride two miles to avoid transferring buses but does not want to ride all the way to his destination.

Because it is easy to select enough routes that it may be hard to visually distinguish all selected routes, users can hover over a stop or selected to see a tooltip that displays the routes or route involved. Note that it still maybe useful to see many selected routes together, such as when visualizing all routes that serve a neighborhood.

Often, there are times of the day during which bus routes do not run. Users can move the slider at the top of the

screen to reveal which of the selected routes run at certain times, as shown in figure 6. The slider runs from midnight to midnight and can be set at half-hour intervals. By default, the slider is set to show routes with trips between 8am and 10pm.

Finally, double-clicking on a route selects all routes with which that route intersects. This is most useful for routes in areas less well-served by mass transit than Downtown Seattle—such as suburban areas—as this helps transit riders visualize where they could go with one transfer. This may particularly important for potential transit riders who do not have drivers’ licenses and may not even look for routes to a destination if they do not already know of a way to get there via mass transit. The feature is less useful for routes in areas well-served by mass transit, such as Downtown Seattle, because selecting intersecting routes through these areas selects most of the routes in Seattle.

## 5. DISCUSSION

Although this work largely accomplished our initial goals, we encountered some significant challenges that will impede future work.

### 5.1 Data

GTFS is a standard format for transit data and includes fields specifically for drawing routes. Each trip for each route is paired with a series of ordered coordinates for drawing the route. Due to the ordering of the coordinates, if data from *all* routes are used, the routes drawn are wildly incorrect. To solve this problem, we draw coordinates from only one trip per route, but this means that any route that does not follow exactly the same streets in both directions will be missing data on our map. Additionally, some routes follow slightly different streets on weekends or holidays, while our map is agnostic of the day of the week.

### 5.2 Bells and Whistles

Another challenge was visually encoding information without adding lots of text and buttons. There are a limited number of user interactions that do not involve adding buttons and text input boxes and because of the breadth and depth of information we wanted to visualize, we had be judicious about adding features. For example, it would have been easy to implement the time-of-day filter as two text input boxes, but we feel that a slider conveys as much information as the text boxes would have and is more intuitive to users, as it is simple to move around and encourages the user to try different times without explicitly typing the numbers.

### 5.3 Overlapping Routes

Bus routes frequently overlap, an extreme example being all the routes that use Interstate 5 to travel over the Lake Washington Ship Canal. In our representation of transit data, overlapping routes are simply drawn on top of each other, defaulting to the order in which d3 receives them. Although we did not successfully implement a solution that draws overlapping routes side-by-side, as this would involve very specifically changing the actual pixels or coordinates in the data structures that hold the routes, we did automatically draw routes that the user hovers over on top of all other routes, which makes it significantly simpler for a user to explore overlapping routes.

## 6. FUTURE WORK

Because of the richness of the data, there are many potential extensions to this work involving schedules or timing. However, we do not want to imitate existing works for point-to-point navigation ([7]) or real-time navigation ([2]). To this end, future work could focus incorporating timing data in the following ways:

1. Show users how far they can travel in a given amount of time from a bus stop. This could be done by decreasing the saturation of the color of the selected routes proportionally to the time the bus is scheduled to take to get to that point on the route.
2. Show frequency of routes. We considered doing this by using animated dots or bus icons that move along the route to display the number of buses running at a given point in time. Another way to show this could be to make the thickness of the line proportional to the frequency.
3. Show any significant gaps in service. For example, this would indicate commuter routes that only run in the morning and evening, or night owl routes that run late at night.
4. Show typical lateness along a given route in a time frame. This data would have to be learned over time and would be subject to other influences that would not be displayed in the data.

Additionally, we would like to add more features to the sidebar to provide additional information or make interactions easier.

1. Show OneBusAway real-time arrival data directly in the sidebar when stops are selected.
2. Add buttons to deselect routes from the sidebar.
3. When hovering over the route number, highlight the route in the map.

Lastly, we would like to make this visualization available for other mass transit systems. Many transit agencies have released their route and schedule data in the GTFS format, and our visualization is mostly agnostic to location, so it would just be a matter of processing the data and making it available to the client. In the long term, we could set up a server that regularly pulls new GTFS data from transit agencies across the world, and automatically processes it so that AwesomeBus always has fresh data.

## 7. CONCLUSION

We have created AwesomeBus, an interactive visualization of public transportation in Seattle. AwesomeBus visualizes the entire transit network and provides multiple interaction techniques to select, filter, and query transit routes and stops. These features allow users to better understand which bus routes are available to them, and which destinations are easily accessible by bus. Compared to existing visualizations, AwesomeBus does not provide trip planning or real time data functionality, but makes it easier to understand the system as a whole by showing accurate geography and multiple routes at once. We believe that our approach can help people explore the possibilities of public transportation, and encourage greater utilization of mass transit into the future.

## 8. REFERENCES

- [1] 2016. Dots on the Bus. (7 June 2016). <https://urban-data.herokuapp.com/>.
- [2] 2016. OneBusAway. (7 June 2016). <http://onebusaway.org/>.
- [3] 2016. Transit Quality and Equity. (7 June 2016). <http://ray-mon.com/urbandatachallenge/>.
- [4] Srinivas Ashok. 2016. Frustration Index. (7 June 2016). [frustration-index.herokuapp.com](http://frustration-index.herokuapp.com).
- [5] Mike Barry. 2014. Visualizing MBTA Data: An interactive exploration of Boston’s subway system. (10 June 2014). [mibtaviz.github.io](http://mibtaviz.github.io).
- [6] Howard Butler. 2008. GeoJSON. (16 June 2008). <http://geojson.org/>.
- [7] Google. 2016. Google Maps. (2016). [maps.google.com](http://maps.google.com).
- [8] King County Metro. 2016a. Ridership - Monthly Performance Measures. (2016). <http://metro.kingcounty.gov/am/reports/monthly-measures/ridership.html>.
- [9] King County Metro. 2016b. Schedules & Maps. (2016). <http://kingcounty.gov/depts/transportation/metro/schedules-maps.aspx>.
- [10] Ariel Schwartz. 2013. Urban Data Challenge. (19 April 2013). <http://www.fastcoexist.com/1681868/3-creative-ways-to-visualize-urban-public-transportation>.
- [11] Sound Transit. 2016. Schedules. (2016). <http://www.soundtransit.org/schedule>.
- [12] Oran Viriyincy. 2016a. The Seattle Transit Map. (28 April 2016). <https://seattletransitmap.com/view/>.
- [13] Oran Viriyincy. 2016b. The Seattle Transit Map and Guide. (7 January 2016). <http://seattletransitblog.com/2016/01/07/the-seattle-transit-map-and-guide/>.