PAPER No. 14-0840

A VISUAL APPROACH TO PROVIDING BUS ARRIVAL PREDICTION INFORMATION

By

Juan Argote
(Corresponding author)
Department of Civil and Environmental Engineering
University of California, Berkeley
416D McLaughlin Hall, Berkeley, CA 94720
Email: juan.argote@berkeley.edu

Jean C. Doig
Department of Civil and Environmental Engineering
University of California, Berkeley
416D McLaughlin Hall, Berkeley, CA 94720
Email: jcdoig@berkeley.edu

Transportation Research Board
93rd Annual Meeting
Washington, D.C.
January 12-16, 2014

Word Count: 3,663
Figures/Tables: 4
Total: 4,663

July 20, 2013

ABSTRACT

Passenger Information Systems have gained popularity in the past decade and have become a de facto standard in the transit industry. The information provided by these systems improves the usability of transit and provides a better overall experience for its users. However, the interfaces used to present real-time information are generally static and complicated to navigate, while the information that they present should be dynamic and easily accessible.

Our goal is to provide a single, dynamic, and intuitive interface where transit users can easily infer the next bus arrival for any stop along a particular route. To do so, we present a web application combining the reported position of the buses and the arrival prediction information to color code arrival estimations along the entire route. The passenger only needs to glance at the color surrounding her stop to infer the time until the next bus arrival. This visualization also uses geo-fencing to suggest which stop to access. A version of the visualization that uses historical data can be accessed at "*www.vudlab.com/beartransit.html*".

INTRODUCTION

Passenger Information Systems (PIS) have become widespread among transit agencies around the world. Access to real-time location enables these systems to provide valuable information to transit users and operators alike (*1*). In particular, transit commuters and occasional riders can access real-time information to improve their decisions when planning a trip. However, in some instances, the size and complexity of public transportation systems in combination with a careless design of the information system's interfaces may constitute a barrier.

This paper presents a new visual approach to display real-time information in a way that can be interpreted and understood easily by both transit users and operators. Particular emphasis is given to the visual display of next arrival predictions for transit users, but this approach can be used to report other information concerning the state and operation of the route. Next arrival predictions are highly demanded by public transport users (*2*), especially in routes with large headways, making its visual representation a useful application.

PIS generally present arrival predictions numerically. However, accessing this information can be cumbersome. Most PIS require users to know and input not only their desired route and direction but also the specific stop at which they plan to board. Moreover, some PIS' also require the user to know the specific stop ID, information that may not be available to the user until physically reaching that particular stop.

Our proposed visualization intends to diminish the number of steps required to retrieve arrival information at a stop. We use a single graphical interface where the route map itself conveys arrival information in real-time. Each span of the route is colored to convey the time until the next estimated arrival. Figure 1 shows an implementation of our visual approach using real data from the Perimeter route of the Bear Transit campus shuttle service, at UC Berkeley.

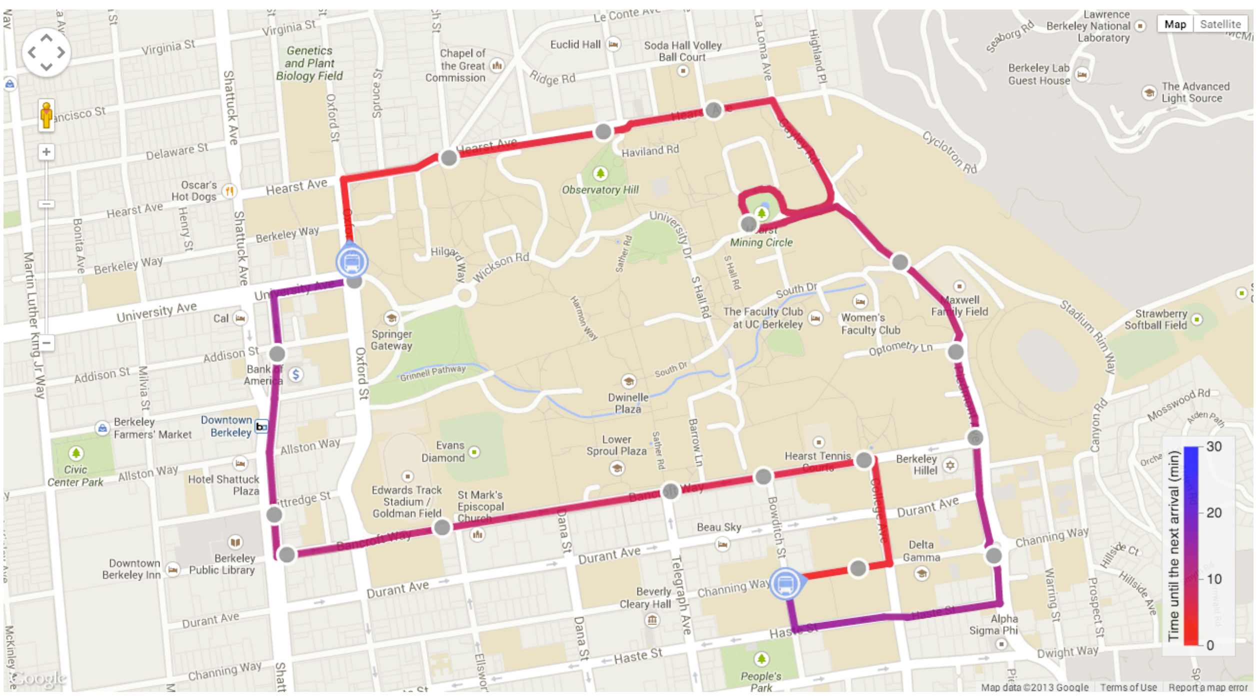


FIGURE 1 Next arrival predictions visualized on top of the route

In the following section we discuss the benefits of PIS and review some of the existing systems. We thereafter present the system architecture, data collection, and data processing required to generate the proposed visual representation. The visual interface and some extra functionalities are then described in detail. Finally, we present our conclusions.

EXISTING PASSENGER INFORMATION SYSTEMS

Real-time transit PIS have become a de facto standard in the public transportation industry. A significant number of transit agencies currently provide real-time arrival information to the public. For example, in a recent survey conducted in (*3*), 89% of the respondent transit agencies reported having real-time prediction software. These systems provide a variety of benefits to the passengers and improve the usability of transit. The information facilitated can supplement the users' decision-making ability when planning a trip and at the same time provide a better overall experience (*1*). For example, users waiting at a stop served by multiple routes may adapt their trip based on the route of the next bus to arrive.

Moreover, this information can help mitigate the negative impact of extended and unanticipated wait times (*4*, *5*). If the PIS provides reliable information, then uncertainty over wait time decreases. If headways are large enough, knowledge of wait times might even allow passengers to use their wait time at their convenience in the area surrounding a stop.

Multiple studies have assessed and quantified the benefits of real-time transit PIS. While initial studies found that real-time information systems would provide small benefits to the users (*6*), more recent literature has identified significant benefits for both passengers and transit agencies (*7*, *8*, *9*). This is attested by the rapid rate of adoption of this technology.

However, there is limited literature concerning the design and implementation of interfaces to communicate this real-time information. An early attempt to provide the guidelines for such design can be found in (*10*). It includes a thorough taxonomy of the functionalities and contents that a transit PIS should include but it only provides a basic example on how a system would provide information on the real-time position of the buses. More recent efforts, like the one presented in (*11*), include several features that facilitate real-time information to the users and have established a trend towards the development of more usable and attractive ways of delivering real-time transit data.

In fact, numerous agencies and AVL providers are already following this trend. For example, NextBus (*12*) has recently added a feature that plots the location of buses in real-time and allows the user to retrieve real-time information at a stop by clicking on it. Other examples of applications that allow to track the position of the buses with real-time data feeds are (*13*, *14*). However, none of these examples update bus locations frequently enough to create a very responsive and dynamic visualization.

The data for a particular bus seems to be updated very infrequently (every few minutes) and buses remain static at their latest reported position on the map. Moreover, even though the users can view the position of the buses within the route, it may be still quite problematic to infer the wait time at a stop without navigating to some dedicated interface, where the predicted wait time is explicitly presented. In view of this shortcoming, we have developed a visualization framework that allows the user to infer the time until the next arrival at each point along a route without navigating away from the map interface.

DATA COLLECTION AND PROCESSING

The visualization approach presented in this manuscript requires both static and dynamic data. The static data, which in this case follows the General Transit Feed Specification format (*15*, *16*), must include information about the shape of the route, the location of the stops, and the underlying published schedule. The dynamic data must at least report the position of the buses in real-time. It is generated by AVL systems (*17*), which equip buses with GPS devices, match the buses with specific trips in the underlying service schedule, and transmit their position at a certain frequency.

These data, alone, are not sufficient to provide real-time arrival predictions. The data need to be fed to an intermediate server that predicts arrival times. In our particular application, we have resorted to the open source application OneBusAway (*11*), which provides all the necessary features to project delays downstream and generate arrival predictions for all stops on a route for a given bus trip. FIGURE 2 presents a diagram of the system architecture.

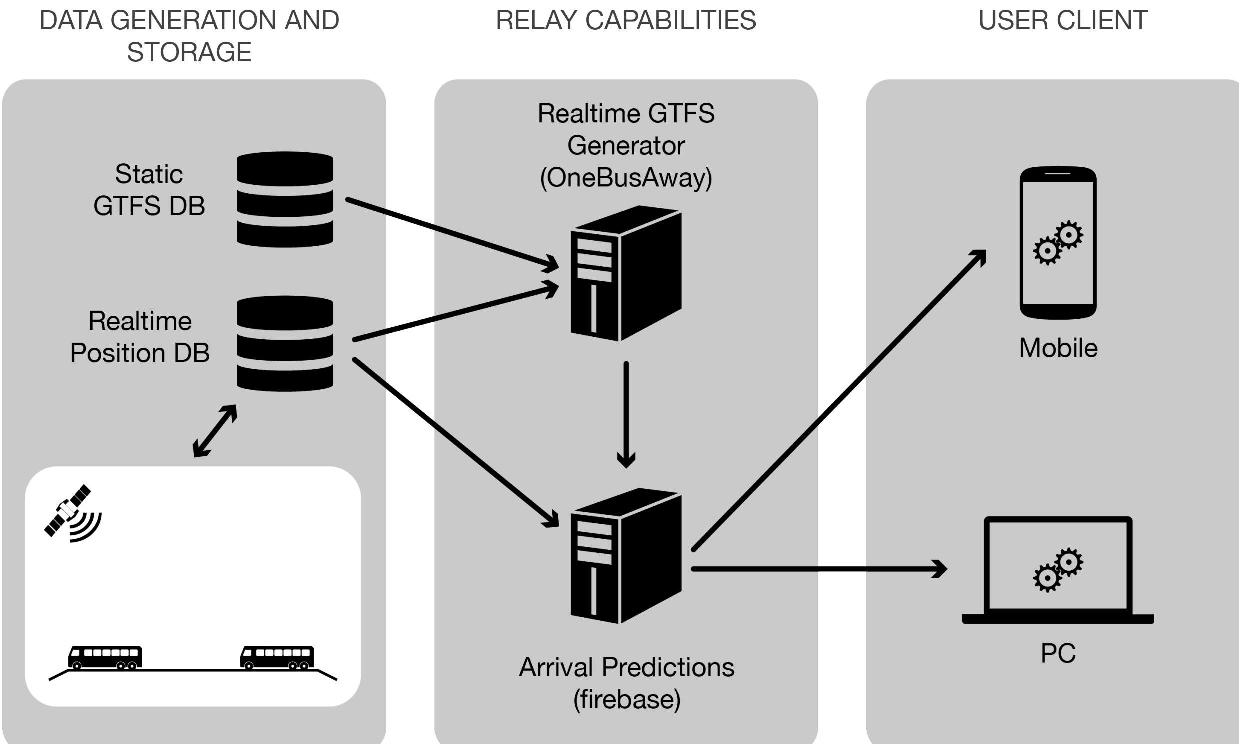
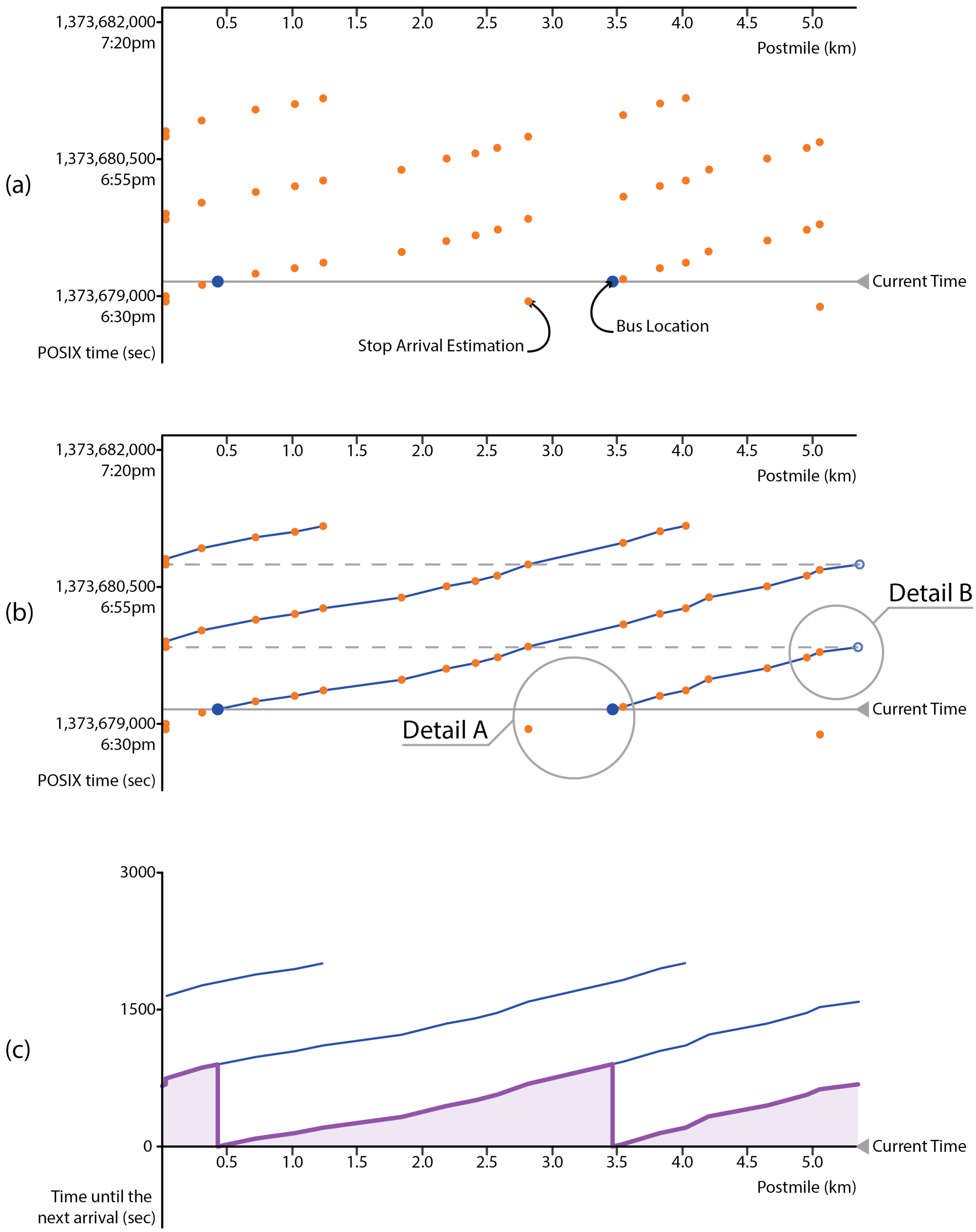


FIGURE **2** System Architecture

Once the static and dynamic data have been jointly processed by an intermediary server, the arrival prediction and bus position information are ready to be delivered to the final user client. However, these data still need to be modified by a web application in order to create a dynamic visualization.

Client side data processing

The user client retrieves predictions from the web server for every active trip on a route. As described in (*15*), a trip is “a sequence of two or more stops covered by the same vehicle that occurs at a specific time.” Relevant data for each trip include the arrival estimation at each stop and the current location of the bus serving the route. However, bus location might not be reported if the trip has not yet started.



1
2 **FIGURE 3 Data Processing as performed by the user client: (a) data points received from the**
3 **server, (b) trajectories, (c) next arrival function**

FIGURE 3a shows a space-time diagram depicting every data point received from the server at a given time for all active trips. Note that these points can be used to reconstruct the predicted bus trajectories by connecting the data points generated by a particular bus. The server asynchronously updates each of these data points as new data arrive from the buses. By ‘asynchronously,’ it is meant that the web server can update its predictions or bus locations at any time and in any order. Whenever a data point is updated, the web application can use it to recalculate a trajectory. For the sake of simplicity, the trajectories are calculated by connecting the data points using linear interpolation, as shown in Figure 3b. It should be noted that the slope of these trajectories represents the predicted pace of the bus.

It is also important to note that the asynchronous nature of the updates impacts how the new trajectories are calculated. It is possible that, for a brief period of time, a portion of the old data used to calculate the new trajectory could contradict the newly received data.

The user client uses the data points as soon as they are received. However, it needs to correct any inconsistencies before presenting the information to the user. Potential problems and the logic that the client follows to solve them is described in the following paragraphs. The advantages of asynchronous updating are smaller data packages and more frequent updates.

Processing bus location updates

Bus location updates provide the latest reported location of a bus. They are essential, because they reflect real conditions and do not involve prediction. Therefore, at any given time the trajectory of a bus should begin at the reported bus’ location.

A possible conflict arises when a bus passes a stop where the arrival predictions have not yet been updated. In this case, the prediction might indicate that the bus still hasn’t arrived. To solve this issue, we will ignore the arrival predictions of stops upstream from the reported bus position when constructing the piecewise linear trip trajectory. Figure 3b Detail A illustrates this logic. Note that even if the data point is consistent with a forward moving trajectory (the expired arrival prediction lies below the current time), ignoring it will not cause problems.

Processing stop arrival predictions update

A new arrival prediction reports a change in the expected wait time for one or more stops. This might occur in response to changes in traffic conditions or other variables that could affect a bus’ motion. The new arrival predictions will be used to reconstruct the trips’ trajectories by joining the new data points using a piecewise linear function. In this case, only the time coordinate for the set of updated stops will vary, affecting the slope of the piecewise linear function.

This procedure may prove insufficient for looping bus routes. On loops, the last stop of one trip is also the first stop of the next one. Thus, to guarantee consistency the arrival information at the last stop must be duplicated. FIGURE 3b Detail B shows this process: a new data point (displayed as a blue circle) is added to the trajectory at a distance from the origin that is equal to the initial stop postmile plus the entire length of the route.

Estimating the next arrival function

The final step is the estimation of the so-called “next arrival function”, $T_R(x,t)$. This function, like the trajectories used to derive it, is piecewise linear. It maps the time left until the next bus arrival at any location along the route. Equation 1 shows the formula used to generate the next arrival estimation function using the trajectories of each active trip i in the route of interest.

$$T_R(x,t) = \min_i(\Delta T_i(x,t)), \tag{1}$$

where,

$$\Delta T_i(x,t) = \begin{cases} \infty & : L_i(x) \leq t \\ L_i(x) - t & : L_i(x) > t \end{cases} . \quad (2)$$

Since buses move forward within a trip, the trajectories can be mathematically represented as monotonically increasing piecewise linear functions $L_i(x)$ (see FIGURE 3b). These functions return the time at which the bus serving trip i will arrive to location x if it follows the current trajectory.

The function $\Delta T_i(x,t)$ denotes the time remaining until the arrival of the bus covering trip i at location x. Notice that this function returns a time difference (from the current time t). If the bus has already moved past a location x at time t, $\Delta T_i(x,t)$ will return infinity.

The next arrival function is the lower envelope of all ΔT_i functions. Note, that this lower envelope depends on the current time t. While trajectories need to be recalculated only when a new data point is received from the server, the next arrival function needs to be frequently updated to achieve a dynamic visualization (we set our updating period to 200 milliseconds). Even if the connection with the server is lost, this function is still updated based on the latest available trip trajectories and the current time. The resulting visualization will display buses that move continuously along the route (with the pace dictated by the trip trajectory), arrival estimations that change with time, and smooth visual transitions. FIGURE 3c shows a snapshot of the next arrival estimation function at a specific time.

VISUAL REPRESENTATIONS OF ARRIVAL PREDICTIONS

Once the piecewise linear arrival prediction function has been determined, it is possible to obtain the expected arrival time for the next bus at each point along the route. The visualization uses these values and maps them to a color gradient so that the entire route shape is colored in a way that represents the time left until the next arrival. The stops are then placed on top of the line at their actual physical location. Similarly, the buses are also depicted and their positions updated frequently, as described in the previous section.

This visualization approach enables the user to obtain arrival information for multiple stops from a single interface while selecting only the route and direction of travel (the latter is only convenient in routes that have segments where a single street is used in both directions). Once this information is given, the color of the line surrounding any stop indicates the time left until the next predicted arrival. A legend located on the bottom right corner of the screen contains the numerical values associated with each color in the gradient. This visual tool eliminates the need to navigate to a different interface where numerical information is displayed in isolation, as is standard in the current PIS. In addition, our interface does not require users to know the name or id of a stop—information that may not be known to infrequent riders.

In addition, users can retrieve a numerical arrival prediction for a specific stop, within the main interface, by clicking on that stop, as shown in FIGURE 4. This action will zoom in on the selected stop and display a text window containing the time remaining until the next arrivals and some additional information such as the name of the stop.

Additional functionalities: dynamic catchment areas

Moreover, when selecting a stop, the user will be presented with a dynamic catchment area surrounding that stop. This feature makes it easy for a user to gauge whether she will reach that stop in time to board the next bus. If the user's position lies within the gray shaded area, see FIGURE 4, then she should be able to comfortably walk to the stop in time to board the next bus. Thus, the catchment area radius at stop s and time t, $r_s(t)$, must depend on the time left until the next arrival:

$$r_s(t) = \frac{T_R(x_s, t)v_w}{\sqrt{2}} \tag{3}$$

where v_w is a representative walking speed (e.g. 3 km/hr in urban environments with frequent street crossings). To be conservative, the product of these two terms is divided by $\sqrt{2}$.

Since the catchment area is circular but the streets may not offer the possibility of walking on a straight line, it is possible that a user within the circle would have to walk up to $\sqrt{2}$ times the distance from his original position to the stop. This correction is important to avoid situations in which the user is encouraged to walk to a stop, only to see the bus vanish into the horizon. As described in (*18*), such negative experiences could drive riders away from transit.

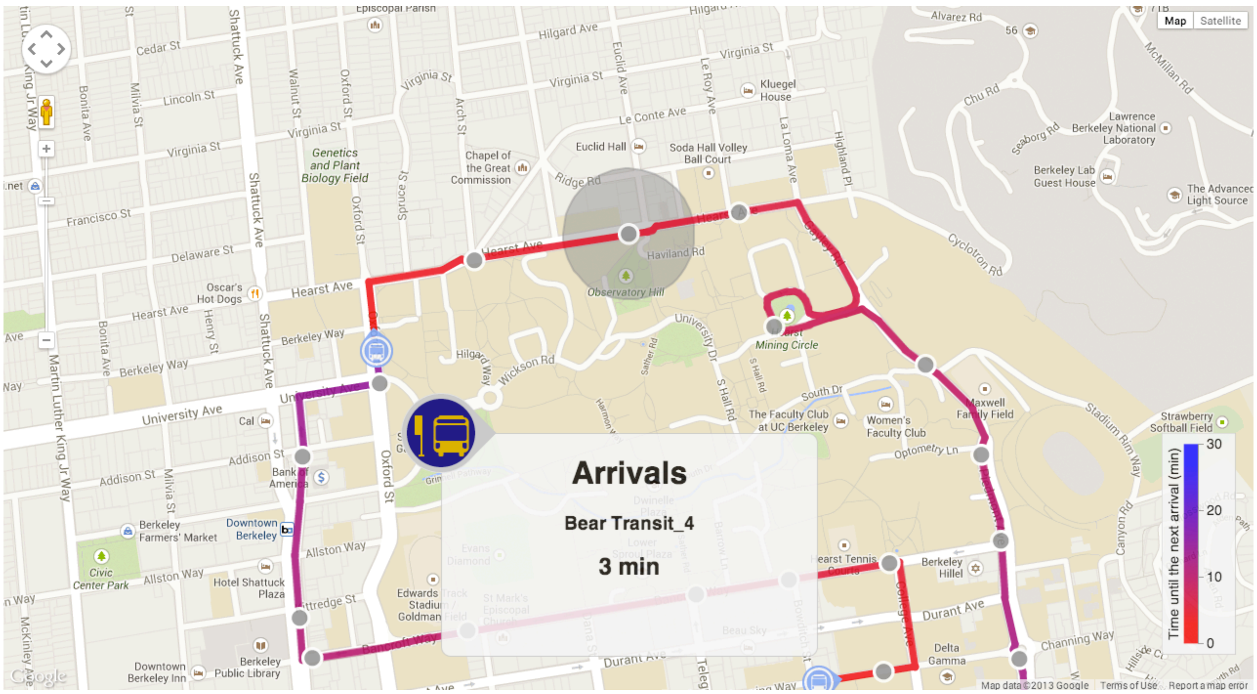


FIGURE 4 Arrival information presented after clicking a stop

CONCLUSIONS

This paper introduces a new visual framework for providing arrival predictions to transit users. The main goal is to reduce the number of inputs necessary to retrieve predictions. In this visualization framework, the riders are only required to know the route and direction of travel. Once users select these two inputs, they are presented with a visual representation of the route that allows them to infer the time left until the next bus arrival at any point along the route.

With this approach, the user can compare which stop is the most convenient for her trip using a single interface. This simplification improves the passenger decision-making process. Moreover, in this visualization we briefly explore the combination of arrival information with geo-fencing. After selecting a particular stop, the user would observe a dynamic catchment area that could tell her about the likelihood of reaching the next stop in time to catch the next bus. In view of that, the user could potentially choose to use a different stop than originally anticipated in order to catch the first available bus.

A primary difference from existing bus visualizations is the update engine designed to run on the client side. This engine allows a dynamic representation of arrival information, flowing with time instead of presenting static and discrete updates, such as those reported by the server. Thus, in the event of a communication loss between the server and the client, this visualization approach would keep updating based on the latest available trajectories, providing the user with a more realistic depiction of the real-time state of the route (if the underlying arrival prediction algorithm is sufficiently accurate).

Future work will seek to improve the visualisation tool with the user geographic information. Current browser and mobile technology allows for the user client to request the current location. This information could be combined with the PIS data to automatically show the closest stops that could satisfy the user's travel needs.

ACKNOWLEDGMENTS

The authors would like to thank the UC Berkeley Bear Transit system and Via Analytics for providing the necessary infrastructure to carry out this visualization project. Finally, the authors also want to express their gratitude to the team who originally developed OneBusAway.

REFERENCES

[1] Mishalani, R., Lee, S., and M.R. McCord. Evaluating Real-Time Bus Arrival Information Systems. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1603*, Transportation Research Board of the National Academies, Washington, D.C., 2000, pp. 81–87.

[2] V. Scinteie. Implementing passenger information, entertainment, and security systems in light rail transit. In *Transportation Research Circular E–C058: 9th National Light Rail Transit Conference*, 2003, pp. 528-533.

[3] Transit Cooperative Research Program. *Synthesis 91: Use and Deployment of Mobile Device Technology for Real-Time Transit Information*. Transportation Research Board, Washington, D.C. 2011.

[4] Bowman, L.A. and M.A. Turnquist. Service frequency, schedule reliability and passenger wait times at transit stops. *Transportation Research Part A: General*, Vol. 15, No. 6, 1981, pp. 465-471.

[5] Mishalani, R.G., McCord, M.M., and J. Wirtz. Passenger wait time perceptions at bus stops: empirical results and impact on evaluating Real-Time Bus Arrival Information. *Journal of Public Transportation*, Vol. 9, No. 2, 2006, pp. 89 – 106.

[6] Hickman, M. and N.H.M. Wilson. Passenger travel time and path choice implications of real-time transit information. *Transportation Research Part C: Emerging Technologies*, Vol. 3, No. 4, 1995, pp. 221 - 226.

[7] Lehtonen M., and R. Kulmala. Benefits of Pilot Implementation of Public Transport Signal Priorities and Real-Time Passenger Information. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1799*, Transportation Research Board of the National Academies, Washington, D.C., 2007, pp. 18-25.

[8] Dziekan, K. and K. Kottenhoff. Dynamic at-stop real-time information displays for public transport: effects on customers. *Transportation Research Part A: Policy and Practice*, Vol. 41, No. 6, 2007, pp. 489 - 501.

[9] Watkins, K.E., Ferris, B., Borning, A., Scott Rutherford, G. and D. Layton. Where Is My Bus? Impact of mobile real-time information on the perceived and actual wait time of transit riders. *Transportation Research Part A: Policy and Practice*, Vol. 45, No. 8, 2011, pp. 839 - 848.

[10] Peng, Z.R., and R. Huang. Design and development of interactive trip planning for web-based transit information systems. *Transportation Research Part C: Emerging Technologies*, Vol. 8, 2000, pp. 409 - 425.

[11] Ferris, B., Watkins, K., and A. Borning. OneBusAway: Results from Providing Real-Time Arrival Information for Public Transit. *CHI '10, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Atlanta, GA, April 2010, pp. 1807-1816.

[12] NextBus. http://www.nextbus.com/googleMap/?a=actransit&r=B&d=B_57_1&s=9902310. Accessed July 2013.

[13] Joe Shaw. Where's the bus? http://mbta-bus.appspot.com/. Accessed July 2013.

[14] Chicago Transit Authority Bus Tracker. CTA Bus Map. http://www.ctabustracker.com/bustime/map/displaymap.jsp. Accessed July 2013.

[15] Google. General Transit Feed Specification. https://developers.google.com/transit/gtfs/reference. Accessed July 2013.

[16] GTFS Data Exchange. http://www.gtfs-data-exchange.com/. Accessed July 2013.

[17] M. Hickman. Bus automatic vehicle location (AVL) systems. *Assessing the Benefits and Costs of ITS*, Ed. Gillen D. and D. Levinson, Springer, 2004, pp. 59 – 88.

[18] Carrel, A., Halvorsen, A., and J.L. Walker. Passengers' Perception of and Behavioral Adaptation to Unreliability in Public Transportation. Presented at 92nd Annual Meeting Transportation Research Board, Washington, D.C, 2013.