# Bus-Lambda: A low-cost, real time transit information system

Varun Adibhatla[1], Aaron D'souza[1], Xia Wang[1], Vishwajeet Shelar[1], and Manrique Vargas[2]

[1]ARGO
[2]NYU CUSP

July 8, 2019

### Abstract

As the world urbanizes and megacities continue to attract vast populations to their economic cores, the bus has remained a vital part of the public transit stack to ensure the efficient and safe movement of people, at scale.

Enhancing and preserving the quality public transit as billions of people emerge out of extreme poverty and migrate to cities in the pursuit for better lives, remains challenging for many transit and municipal entities. It is therefore vital that cities find sustainable methods to improve the quality and safety of the public transit experience with fewer resources.

Real-Time Transit Information Systems is an opportunity that has shown to increases in ridership, substantial increases in fare revenue, reduced waiting times, and overall satisfaction with transit service. Many cities could could achieve these outcomes for millions by reducing friction and lowering the barrier to implementing real-time transit information at scale, sustainably. Encouragingly, with advances in cloud computing, Wifi positioning system, and inexpensive hardware this missing layer of useful information between transit providers and users amounts to a common-sense integration and deployment of technology.

In this paper we offer the design, development, and deployment of Bus-Lambda, a low-cost system to track buses in a city. The system itself leverages advances in low-cost hardware, cloud computing and critically, a means of acquiring a bus' location without the use of a dedicated GPS device.

We argue that Bus-Lambda can address information asymmetry of bus location awareness and significantly reduce the barrier and cost of operating a real-time transit information system towards the goal of a successful, highly scalable implementation. Cost estimates of implementing Bus-Lambda across an entire city's bus network is between $75-$100 per bus compared to current implementations that cost many thousands of dollars.

# Introduction

Despite advances in micro mobility and ride sharing, fixed route buses remain the “best way to get the most people around a city efficiently and cheaply”. (Walker, 2018)

A key facet of an efficient fixed-route bus network involves buses maintaining a fixed schedule at fixed bus stops. However, in rapidly urbanizing cities and megacities across the world, increasing congestion increases uncertainty and inefficiencies in bus operations causing unexpected delays and bus-bunching.

Real-Time transit information systems is one such opportunity. Real-time transit which involves tracking the locations of transit and displaying simple information about When the next bus or train is arriving has shown to reduce waiting times at bus stops by 15 percent compared to commuters without such information. (Watkins, Ferris, Borning, Rutherford, & Layton, 2011) Access to real-time transit information has also been linked to overall satisfaction with transit service, increases in ridership, and substantial increases in fare-box revenue. (Brakewood, Macfarlane, & Watkins, 2015; Tang & Thakuriah, 2012; Zhang, Shen, & Clifton, 2008)

If cities could simply increase the practical availability to real-time transit information through common-sense deployment of technology, they could achieve outcomes similar to increases in transit service itself and avoiding a more resource intensive effort.

Encouragingly, with advances in cloud computing and inexpensive hardware, this missing layer of coordination between transit providers and users amounts to a conceptually simple piece of technology. Less encouragingly, legacy technology and a procurement process that prefers complexity often leads to exorbitantly expensive, over-engineered complex systems that create delays and other barriers to service delivery.

NYC’s real-time transit information system, for example, which includes the Metropolitan Transportation Authority’s (MTA) BusTime mobile app costs many thousands of dollars per bus.(Cubic-Transportation-Systems, 2014) In a system that spans well over five thousand buses, the operating costs to simply track a bus’s location every 30 seconds can seem difficult to fathom.

Granted that the costs include redundancy and fault-tolerance to ensure that the system operates at all times in a reliable manner.

After a protracted 3 year deployment process, Melbourne, a moderately-sized city, was forced to suspend rollout of its original bus-tracking system in 2013 at only 30% coverage due to unreasonable operating costs.(Coyne, 2014)

To benefit urban residents at massive scales however these implementation and operating costs of real-time transit information are simply not sustainable and must be lowered and offered as an open-source system in the public’s interest. It is with these motivations and objectives that we present Bus Lambda so that transit agencies across the world are empowered to implement effective real-time transit-information systems with minimal resources.

# Key Concepts

## Wifi Positioning

Wifi Positioning was pioneered by a U.S.based firm named Skyhook Wireless in 2007. The system builds a database by passively collecting and integrating uniquely identified wireless hotspots with high-precision location information. This is typically within a dense area such as a city. Skyhook Wireless then leverages this database to provide a coarse-grained location information without requiring dedicated Global Positioning System (GPS) hardware. The system was intended for use by early generation smartphone devices and was a key feature in the deployment of the the first version of the Apple iPhone as a means to preserve battery life while requesting its geographical location. Other terms used to describe this technology are Assisted-GPS or Augmented-GPS (A-GPS). Due to the proliferation of smartphones across the world, many companies including Skyhook, Unwired Labs, and Combain Technologies maintain updated databases of geocoded wifi hotspots across the world at low costs. (Wikipedia, 2019)

Finally, A significant advantage to using Wifi positioning over GPS is avoiding the Urban Canyon effect that causes GPS systems to malfunction in urban areas with a presence of densely packed skyscrapers.

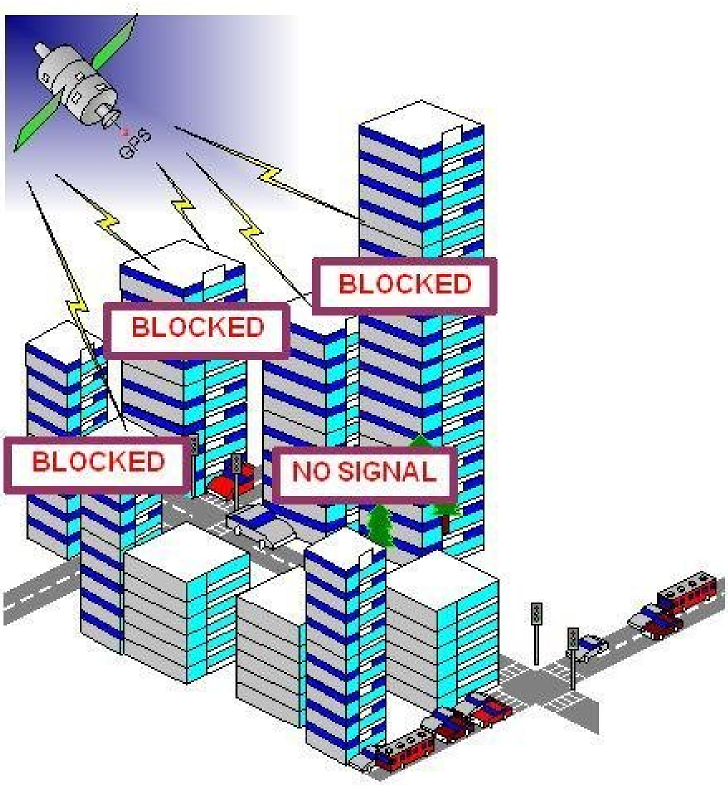


Figure 1: An illustration of the Urban Canyon effect that adversely effects GPS systems. (Byon, 2011)

## Serverless Computing

Serverless Computing refers to a form of cloud-based utility computing where pricing is structured based on the exact resources used by a single or series of applications as opposed to a pricing structure where the entire system needs to be purchased prior to executing the application. This form of computing was introduced to the public at large by Amazon Web Services in 2014 through this Lambda product offering. (Services, 2019) Key advantages of serverless computing include cost and scalability. The setup costs typically associated with enterprise applications is eliminated in a serverless paradigm as hardware is pre-provisioned by the cloud computing provider. This significantly reduces the lifecycle of deploying enterprise grade software and reduces the resultant maintenance debt as thereby reducing overall system complexity. Furthermore, serverless applications are designed to elastically scale to meet incoming demand. Every initialization of a serverless application spawns a separate serverless instances and these instances can scale across the provider's hardware offering ensuring optimal allocation of system resources. A limitation of serverless computing is that they are best used for single purpose computing i.e. performing a small set of distinct tasks such as counting, simple arithmetic, read or write operations.

## General Transit Feed Specification

General Transit Feed Specification (GTFS) is a transit-specific data specification that allows public transit agencies to publish their transit data in a format usable by a wide array of software applications. It is used across the world by many of public transport providers. The GTFS protocol was initially developed in 2005 as a result of a collaboration between Tri-County Metropolitan Transportation (Trimet), the Portland metro area public transit provider, and a  Google employed engineer. (McHugh, 2013)

# Related Work

## OneBusAway

OneBusAway is an open-source software system that implements a real-time transit information system using the GTFS data specification for the basic description of the transit system. OneBusAway was initially developed in 2008 by Brian Ferris and Kari Watkinsat the University of Washington and saw its initial deployment in the Puget Sound area in Washington. Since its initial deployment OneBusAway has since been deployed to other cities including Atlanta, San Diego, Tampa, Washington, D.C., with international deployments in York (Canada), Finland, and Poland.

In 2013, the MTA in New York City implemented OneBusAway software as a core part of its Bus Time system. (Ferris, Watkins, & Borning, 2010; TransitWiki, 2017)

## Swifty

Swiftly is a U.S. based company founded in 2014 by Jonny Simkin, Michael Smith, and Will Dayton that implements an end-to-end real time transit information system. Swiftly's approach includes a platform for several stakeholders including planners, customer service, schedulers, and core bus operations.

The Swifty platform aims to integrate with existing Computer Aided Dispatch (CAD) and Automatic Vehicle Location systems to improve "transit system performance, service reliability, and real-time passenger information". (Swiftly, 2019)

## TransLoc

Transloc is a US based firm founded in 2004 that was acquired in 2018 by Ford Smart Mobility Systems LLC (Etherington, 2018). TransLoc implements real-time tracking primarily for microtransit networks that serve customers in smaller, private networks such as university and corporate campuses. Additionally, the Transloc ecosystem includes demand modeling and response analysis to enhance their mobile application in addition to real-time tracking. The implementation typically includes dedicated GPS hardware.

## Bus-Lambda System Architecture

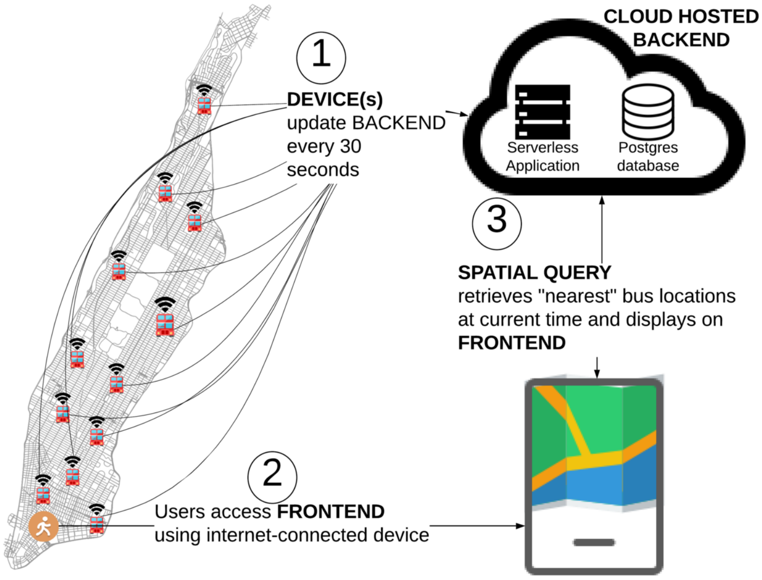


Figure 2: System Architecture

### Device

The Raspberry Pi Model Zero W is a $10 microcomputer that can run the open source Linux operating system. The “W” implies included wireless functionality through a Wifi and Bluetooth adapters that allow this version of the device to connect wirelessly to bluetooth and wifi peripherals. In this deployment, a single device is allocated to each bus is part of the transit network. We envision the device be included as a kit with the following components towards a complete deployment to ensure uninterrupted power and internet connectivity.

1. A micro-SD memory card.
2. A power source such as a solar-powered and charged USB battery kit.
3. A mobile internet hotspot.

The MAC-address of the wireless adapter is used as a unique identifier of each device and can be integrated with fixed-route information to uniquely identify individual buses or the routes they operate on. While it is not required for proper functioning, the solar-powered battery requires access to a source of solar energy which requires the kit to be placed securely on top of the bus facing the sun.

### Frontend

The front end of this system is composed of a single web-page application that can run on any browser compatible, internet connected device. This web-page when loaded connects to the serverless application which queries the database that is hosted on the the cloud provider. The web-page displays a requester’s current location. In this version of the application, we use the Geolocation feature of Google Maps’ Javascript API. (Google-Maps, 2019) This application can be hosted on the widely popular code hosting site, Github through its product offering, Github Pages. (Inc., 2019). A hosted version of the frontend can be accessed at `https://vr00n.github.io/Bus-Lambda`. Please be informed that upon clicking this web link, your browser will prompt you to allow accessing your current location.


Figure 3: Raspberry Pi Model Zero W

## Cloud Hosted Backend

The backend is hosted on a public cloud provider and composed of 3 components:

1. **A Postgres relational database** with PostGIS extension enable containing pre-defined tables to receive 4 data points (timestamp, device identification, route identification, and location coordinates) from individual devices every 30 seconds.
2. **A serverless application** written in the Python Programming language according to the template required by Amazon Web Services' Lambda framework has been written to issue a spatial query (Adibhatla, 2019) to the Postgres database to retrieve a list of devices within a certain distance from the requester's current location as defined by the frontend application at a specific time.
3. **A REST endpoint** that serves as a uniquely identifiable address on the internet to identify the specific serverless application to the frontend application.

# Results

The authors have built a working prototype to demonstrate an end-to-end functioning of the components discussed in the System Architecture section. A real-world deployment was performed on a bus-route in New York City where a device transmitted its location to the backend. A user initialized the front-end at a stop further ahead on the bus route to display the device's recorded location on a map at the specified time.

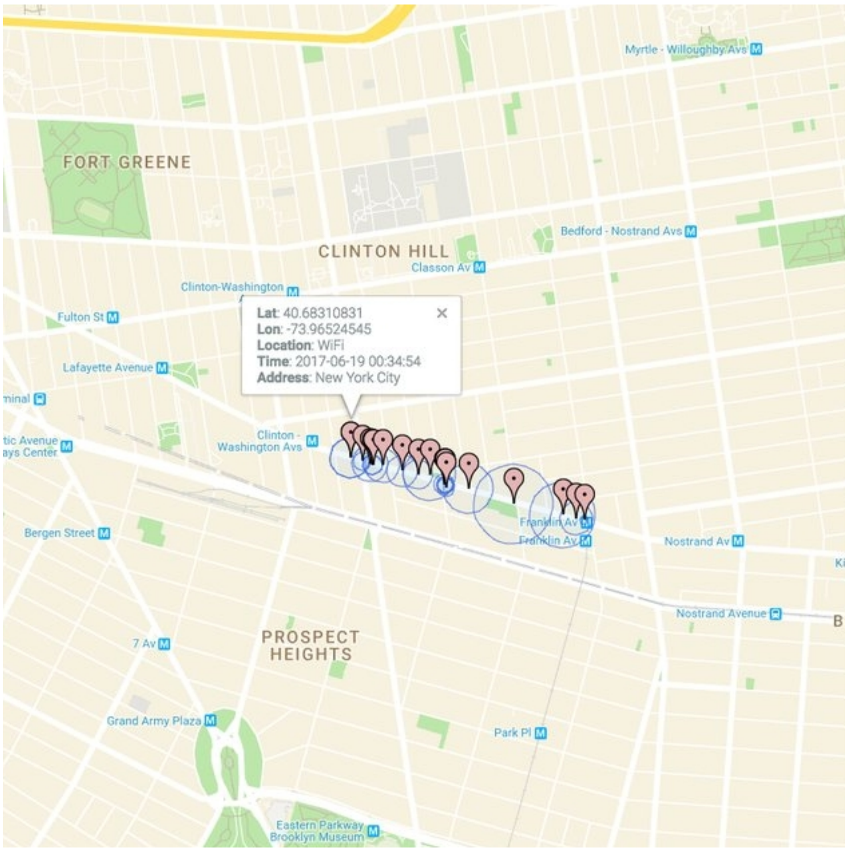


Figure 4: A real-world deployment of the Bus-Lambda system on a bus route in Brooklyn, New York demonstrating the accuracy of Wifi positioning.

## Source of savings

Bus-Lambda offers significant savings over current implementations of network-wide real-time transit information systems in the following areas:

- **Software Setup** - Many orders of magnitude less expensive as Bus-Lambda relies on cloud hosted serverless application architecture.
- **Hardware costs** - Many orders of magnitude less expensive as the system relies on a single Raspberry Pi Zero "W" device which, as of this publication, costs \$10 per unit. The power source and mobile internet hotspot cost \$50 and \$40 in New York City, USA, respectively.
- **Maintenance costs -** Many orders of magnitude less expensive as the entire system, once deployed, can be supported by a single full stack software engineer. Furthermore, individual devices can be easily configured and replaced because of their low cost and simple configurability.

## Future Work

Bus-Lambda's objective is to empower any surface transit agency across the world to implement a low-cost, scalable, and easy to maintain real-time transit information system to significantly enhance public transit ridership, safety, and quality of experience. Future work of the Bus Lambda project includes but not limited to the following core issues and can be tracked at (et. al., 2019) :

1. Integration with the GTFS transit data specification.
2. Incorporating contactless payment systems.
3. Enhancing integration with on-site power and internet connectivity sub-systems.
4. Open source licensing of software.
5. Sales and Deployment Strategy.
6. Comprehensive cost analysis of network-wide implementation compared to existing solutions.
7. Low-cost electronic signage systems.

## Acknowledgements

The authors would like to thank David Marulli, Patrick Atwater, Christopher Tull, Lingyi Zhang, Geoff Perrin, Nicola Macchitella, Swati Bahguna, and Sebastien Burgess for their contributions to this project.

## References

Adibhatla. (2019). vr00n/Bus-Lambda. In *Bus-Lambda*. https://github.com/vr00n/Bus-Lambda/blob/master/findDevice_lambda.py. Retrieved from https://github.com/vr00n/Bus-Lambda

Brakewood, C., Macfarlane, G. S., & Watkins, K. (2015). The impact of real-time information on bus ridership in New York City. *Transportation Research Part C: Emerging Technologies*, *53*, 59–75. https://doi.org/10.1016/j.trc.2015.01.021

Byon, Y.-J. (2011). *GISTMARG: GPS and GIS for Traffic Monitoring and Route Guidance — TSpace Repository* (PhD thesis). University of Toronto. Retrieved from https://tspace.library.utoronto.ca/handle/1807/27572

Coyne, A. (2014). Melbourne takes second stab at GPS bus tracking. *IT News (Next Media)*. Retrieved from http://www.itnews.com.au/news/melbourne-takes-second-stab-at-gps-bus-tracking-381093

Cubic-Transportation-Systems. (2014). Cubic Transportation Systems Adds Winning Technology Innovation to MTA Bus Time(R) for ITS-New York Project of the Year. *Globe Newswire*. Retrieved from http://www.globenewswire.com/news-release/2014/06/16/644279/10085830/en/Cubic-Transportation-Systems-Adds-Winning-Technology-Innovation-to-MTA-Bus-Time-R-for-ITS-New-York-Project-of-the-Year.html

Etherington, D. (2018). Ford acquires Autonomic and TransLoc as it evolves its mobility business – TechCrunch. *Techcrunch*. Retrieved from http://social.techcrunch.com/2018/01/25/ford-acquires-autonomic-and-transloc-as-it-evolves-its-mobility-business/

Ferris, B., Watkins, K., & Borning, A. (2010). OneBusAway: A Transit Traveler Information System. In *Lecture Notes of the Institute for Computer Sciences Social Informatics and Telecommunications Engineering* (pp. 92–106). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-12607-9_7

Google-Maps. (2019). *Geolocation: Displaying User or Device Position on Maps — Maps JavaScript API — Google Developers*. Retrieved from https://developers.google.com/maps/documentation/javascript/geolocation

Inc., G. (2019). GitHub-Pages. Retrieved from https://pages.github.com/

McHugh, B. (2013). Pioneering Open Data Standards: The GTFS Story. *Beyond Transparency*. Retrieved from https://beyondtransparency.org/chapters/part-2/pioneering-open-data-standards-the-gtfs-story/

Services, A. W. (2019). AWS Lambda – Serverless Compute - Amazon Web Services. Retrieved from https://aws.amazon.com/lambda/

Swiftly. (2019). Mobility Platform for Transit. Retrieved from https://www.goswift.ly/products

Tang, L., & Thakuriah, P. (V. (2012). Ridership effects of real-time bus information system: A case study in the City of Chicago. *Transportation Research Part C: Emerging Technologies*, *22*, 146–161. https://doi.org/10.1016/j.trc.2012.01.001

TransitWiki. (2017). OneBusAway - TransitWiki. Retrieved from https://www.transitwiki.org/TransitWiki/index.php/OneBusAway

Walker, J. (2018). The Bus Is Still Best. *The Atlantic*. Retrieved from https://www.theatlantic.com/technology/archive/2018/10/bus-best-public-transit-cities/574399/

Watkins, K. E., Ferris, B., Borning, A., Rutherford, G. S., & Layton, D. (2011). 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*, *45*(8), 839–848. https://doi.org/10.1016/j.tra.2011.06.010

Wikipedia. (2019). Wi-Fi positioning system - Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Wi-Fi_positioning_system

Zhang, F., Shen, Q., & Clifton, K. J. (2008). Examination of Traveler Responses to Real-Time Information about Bus Arrivals using Panel Data. *Transportation Research Record: Journal of the Transportation Research Board*, *2082*(1), 107–115. https://doi.org/10.3141/2082-13

et. al., A. (2019). Bus-Lambda. Retrieved from https://github.com/vr00n/Bus-Lambda