MTA Bus Customer Journey-Focused Metrics Overview

General Description

The Metropolitan Transportation Authority (MTA) is a public-benefit corporation responsible for public transportation in the state of New York serving 12 counties in southeastern New York, along with two counties in southwestern Connecticut under contract to the Connecticut Department of Transportation (CDOT). The MTA is the largest transportation agency in North America.

Bus service within New York City is operated by MTA agencies New York City Transit (NYCT) and MTA Bus Company (MTABC).

As part of the Bus Action Plan, the MTA started to release a series of Customer Journey-Focused Metrics to better measure the rider experience. These metrics are:

> **Additional Bus Stop Time (ABST)** is the average added time that customers wait at a stop for a bus, compared with their scheduled wait time. The measure assumes customers arrive at the bus stop uniformly, except for routes with longer headways, where customers arrive more closely aligned to the schedule. ABST is sometimes referred to as Excess Wait Time.
>
> **Additional Travel Time (ATT)** is the average additional time customers spend onboard the bus compared to the schedule. ATT is sometimes referred to as Excess In-Vehicle Travel Time.
>
> **Customer Journey Time Performance (CJTP)** measures the percentage of customers who complete their journey (ABST + ATT) within five minutes of the scheduled time. CJTP is sometimes referred to as Excess Journey Time.

These metrics are made possible by the advent of large-scale automated data sources, which enabled the creation of a full Origin-Destination (OD) ridership model that assigns each passenger a boarding and alighting time. These indicators are widely considered by the transit industry to be an improvement over traditional vehicle-based metrics. By measuring at the passenger level, these measures weigh the impacts of service delays or changes by the number of passengers affected, unlike past bus-level measures. The shift to such metrics thus also provides the opportunity to better manage service.

> These measures are estimated for each individual bus a customer uses in their journey, also known as an unlinked trip, not all buses in their journey combined. For instance, if someone's

daily commute is to take the Q46 from the Kew Gardens–Union Turnpike subway station and transfer at Union Turnpike and Parsons Boulevard for either the Q25 or Q34 to get to 77th Avenue and Parsons Boulevard, they would have to add the separate metrics for the Q46 and for the Q25 or the Q34.

This dataset was published during the first phase of the MTA's commitment to increasing transparency. We continually examine all our published and publishable data with a view to both providing datasets that can be effectively utilized by our customers and the public at large, and to providing regular, automated updates to these datasets efficiently and sustainably. Consequently, this dataset may be restructured and/or combined with other similar datasets in the future.

Data Collection Methodology

Overview

Since ABST, ATT, and CJTP require information about where and when riders enter and exit buses, and NYCT's Automated Fare Collection (AFC) system does not require riders to swipe or tap to exit, a series of calculations is used. The process takes input ridership data from MetroCard swipes/OMNY taps, GPS tracking data from BusTime, and Automated Passenger Counter (APC) data to estimate the actual and scheduled journey time for each unlinked trip, or leg of a trip, on the bus system.

A simplified bus ridership model uses a bus origin-destination (OD) model and scheduled and actual bus movement data to assign each passenger trip to a bus based on scheduled and actual service. ABST and ATT are calculated as the difference in wait and onboard times between these two assignments for each individual trip, and CJTP is the percentage of rider trips with the sum of ABST and ATT less than five minutes. These measures are then aggregated based on bus route and time period.

Methodology

1. Data Inputs

Automated Fare Collection (AFC)

Automated Fare Collection (AFC) data from fareboxes is obtained from the NYCT Office of Management and Budget (OMB) on a daily basis with a five- to seven-day lag. Data for a given day may arrive late; we expect complete data after a seven-day lag. Every night, when buses return to the depot, data is downloaded to the depot computer and then uploaded to the front

office. The transaction file from the uploaded data contains one record for each MetroCard swipe. It includes transaction type, fare media class, value deducted, the vehicle number, the location code taken from the headsign bus operators punch in, which gives bus route and direction, transaction time (rounded to the nearest six minutes), and number of transactions.

There can be data errors if drivers do not correctly enter the proper sign codes for headsigns or if the headsign units malfunction and show garbled destinations. If there are errors, fareboxes encode an entire day's transactions to one direction.

OMNY

A JSON file with all known OMNY transaction data is obtained the following day. The file includes information like Account ID, Token ID, bus number, the coordinates of the bus when the customer tapped their device or card, bus number, GTFS trip ID, destination sign code, and fare type.

Roadcall data

Department of Buses (DOB) roadcall data is available through a web service set up by the Bus Customer Information Systems (CIS) group. Roadcalls occur when a bus breaks down or has some other issue which places the bus out of service during a scheduled trip. Since the feed is only available for the current day, the roadcalls are imported at the end of every day.

UTS Pullout and Crew Assignment data

Crew assignments are also used to help determine the route and trip information. NYCT's Unified Timekeeping System (UTS) is a non-real-time timekeeping system for bus operators, and also provides a record of vehicles assigned to each pullout from a bus depot (trips from the depot to the beginning of the route without passengers). This connects bus number and operator information to the assigned route and trip. When operator farebox login information is inaccurate or unavailable, which sometimes occurs due to mid-trip reliefs, the UTS information is used to check and validate BusTime inferences. Pullout data is obtained daily from UTS.

Bus schedule data

Bus schedule data is provided in the Surface Timetable Interchange Format (STIF), which is imported each pick (bus operators select, or pick, their work assignments, which correspond with schedule changes, about four times a year, and each schedule is known as a pick). For each route, there are up to four separate files for the type of day (Weekdays school open, Weekdays

schools closed, Saturday, and Sunday). There are six STIF file layers: Timetable, Geography, Trip, Event, Sign Code, Control.

From each file, relevant data include bus stops, the coordinate location of stops, scheduled times for arrivals, departures, or transits (stops that are not timepoints), the bus route, the service code for type of day, the operator job pick the schedule is in effect for, the list of bus trips, unique route paths, the sign codes for the bus routes and direction of services, bus operator run information, including primary run number, primary run route, and block number, and the sequence of stops which make up a trip along the route.

BusTime

Location data is obtained using BusTime, the MTA's automatic vehicle location (AVL) system, which was installed on all buses between 2011 and 2014. The system uses on-board GPS and wireless communication systems and provides the position of all buses every 30 seconds to central servers for processing.

BusTime then infers the run and bus route using information bus operators are required to provide when logging on to the farebox when they board the bus, including information such as their pass number, run route, direction, and run number, and the destination sign code that corresponds to the pixelated messages that appear on the bus headsign. The message can contain words (express, limited, ltd, etc.) that indicates whether the bus is a limited or local.

BusTime data is then integrated with map, route, and schedule data, including Route ID and General Transit Feed Specification (GTFS) Trip ID, previous real-time updates, UTS crew assignments, and bus operator run number.

2. Bus Matching

Bus Matching was developed by NYCT's former Data, Research, and Development (DRD) group in 2015 to match data to routes the day afterward and take the high-volume raw ping data and turn them into a clean table with arrival times for each bus at each stop on the correct route.

There may be several transmitted GPS data points with the same location if buses are stuck in traffic, and precise location at a timepoint might not always be possible for fast-moving buses making a quick stop or not stopping at all, since pings are every 30 seconds. Further complications include when the expected stop location is occupied by a parked car or another bus from a different route.

To address these issues, an algorithm was designed to capture the closest GPS transmitted point to each timepoint based on a coordinate-based distance formula. The closest transmitted point is taken as the time the bus is at that location. If the bus is at the same location for multiple 30-second intervals, the earliest time is taken as the arrival time and the latest time is taken as the departure time.

In addition, for the first and last stop of a bus trip, matching is done differently than at the other stops. The time for the origin stop is the last ping within a 1/8-mile buffer of the origin stop location, and the time for the destination stop is the first ping within a 1/8-mile buffer of the destination stop location. This is done to work around buses laying over at the origin or destination for a long time or passing through the destination without stopping if there is nobody getting on or off the bus.

The inputs for the matching process are data from AFC and OMNY transactions, DOB roadcalls, UTS Pullouts and Crew Assignments, Bus Time GPS with information driver entered into the farebox (trip data) and inference data, and STIF files.

UTS Pullouts Bus Match

After some pre-processing, the procedure determines the scheduled bus number from UTS for every revenue trip for the given route based on schedule day type. It determines all distinct block numbers that contain at least one trip on the given route to be analyzed for the given schedule number (day type and pick).

Trip records for pullout trips where the pullout occurs on the same calendar day are taken from a table defining bus movements based on the STIF Trip Layer. These records are matched with UTS pullouts based on the run route, run number, and pullout time. This process will eventually match on the depot as well, since these three fields are only unique within a depot. After this, each block number is associated with a bus number. Note this bus number may be null or have text indicating a driver did not show up to work or a bus was not available.

BusTime and scheduled trip data

After BusTime data is brought in for a given route, the final step before stop matching is to bring in the bus schedule data in STIF files. This data is joined with the matched block

numbers and the bus number to determine the expected bus for each scheduled timepoint in the dataset.

Stop matching

The goal of the stop matching procedure is to join the BusTime GPS data with the schedule data from STIF. The procedure starts with all the timepoint records and attempts to match to BusTime data given the join conditions described below. It processes BusTime data into arrival and departure times by stop location.

Schedule timepoint records are first matched to BusTime data if the initial join conditions, mainly that there were matches between certain data, were fulfilled. These first matches include records with matches between the actual bus number and the bus number from UTS, matches between the scheduled destination sign code and the inferred sign code, matches between the scheduled route run and the inferred run id, with timepoint locations within 1/8 mile of the bus location, and differences between scheduled time and actual time between –20 and +40 minutes.

Next, records were matched if the previous join conditions were met except that the scheduled route and run and inferred run did not match, in which case, the BusTime data with a potential error is overridden with information from UTS. Records that meet all of the initial join conditions except that the scheduled bus number does not match the actual bus number are then matched using BusTime inference for Destination Sign Code and run id.

The process also matches extra buses to a scheduled route. The process does not match records when the bus is potentially not in service but still traveling along the specified route, with the same conditions as the third join condition, except the phase is “Deadhead_During.” This includes when bus operators put “Not in Service” or “Next Bus Please” on their headsigns during a trip.

During each phase of the matching process, the actual BusTime record with minimum distance is the one chosen. This does not always produce accurate results, especially at origins and destinations where the bus sits in one place for extended periods of time. For origins and intermediate timepoints, the maximum time is chosen as the departure

time, whereas for the destination, the minimum time is chosen. At this point, any unmatched timepoint records are counted as “No Observation”.

Post-processing

UTS crew assignments are downloaded daily and joined to the final matched BusTime data based on run route, run number, and depot. Run number is the “effective” run number, which may change during the trip, for example if there is a mid-trip relief.

If a bus broke down and did not reach a timepoint, it counts as Not in Service (NIS) at all the timepoints it did not reach. Based on UTS data, when a bus did not pull out from the depot due to a driver not showing up to work, equipment failures or unscheduled service changes, the trip is counted as NIS. Buses without hardware or with broken hardware for BusTime can be identified by comparing the BusTime Hardware Installation API and UTS data, which provides bus numbers regardless of BusTime status.

If data are missing, the scheduled time is used as a stand-in for the actual time in the processed bus movement data and the bus is flagged.

The matching process attributes stops to trips using scheduled time. Thus, if a bus is late on a very short route, stops may be matched out of order (that is, passengers arriving before they leave). These passengers are excluded from calculation, because it is not possible to determine whether or not they were able to get on this bus.

There are currently certain routes where, due to difficulty in matching BusTime pings to the schedule, the data quality for these routes is too poor to produce accurate metrics. These are the M35, B39, B42, and QM3. We are working on improvements to our BusTime matching process that will hopefully allow us to begin reporting customer metrics for these routes in the future.

3. APC – BusTime Merge

This step takes in BusTime data matched to the bus schedule and combines it with Automatic Passenger Counter (APC) data to create a single table with the Route, Trip ID, Stop ID, and Scheduled Time for every scheduled bus trip for the dates the model is being run for. For trips matched to actual bus data, the table also includes the bus number and the actual arrival and departure times. APC boardings, alightings, and leave load are also included for trips with APC data. APC data started to be used in the process of calculating bus customer metrics in 2022.

APC data

APCs are infrared sensors mounted above each door on a bus that count the number of boardings and alightings at each stop along a trip. Not all buses are equipped with APCs, and this coverage also varies by borough. All new buses are equipped with APCs; coverage should increase over time as old buses are either retired or retrofitted with them.

Each time that doors open and close, an APC record is created, which includes the bus id, number of people boarding and alighting, a timestamp, and a GPS ping for the geolocation. The run number and destination sign codes are taken from APC records.

Aggregating into trips

Each morning, OP gets stop-level data from APCs for the previous day, and an automated procedure aggregates the data into bus trips. Bus trips are defined by the first APC boarding and the last APC alighting. In order to better define comparisons to AFC data, nearby trips are grouped together into blocks. Trips are grouped together into blocks if there are fewer than 25 minutes between the end of one trip and the start of the next trip on a given bus. This is done since riders may board during one trip than alight during the next. Without grouping, both trips are imbalanced, with the first trip having one additional boarding, and the second having an additional alighting.

Until 2024, when fewer buses had APCs, data from routes with less than one percent APC coverage was filtered out. Then, APC records are matched to MTA BusTime stop arrivals data as long as timestamps are within 90 seconds of each other and based on geographic distance. In doing so, multiple door open/close records will be consolidated to a total ons/offs for the stop.

4. AFC – Bus Trip Matching

Next, Automated Fare Collection (AFC) transaction data (MetroCard swipes) and OMNY taps are matched with bus stop arrival time and bus trip data from the first step using AFC transaction time. Because MetroCards and OMNY are only dipped or tapped at entry, only boarding points can be determined.

Based on the time of swipe or tap, and the bus number, each passenger trip is matched to a bus trip and a set of candidate origin stops. For OMNY, there is usually only one potential stop, because tap time is precise, while, for MetroCard, there are multiple origin stop candidates, since the memory limitation of bus fareboxes when MetroCard was introduced constrained time

stamp resolution to every six minutes, during which time the bus might have covered multiple stops. Thus, several possible boarding stops are selected. For Select Bus Service (SBS), we know the stop, but not the bus a rider boarded.

5. AFC Sorting

This step takes out all AFC transactions (both subway and bus trips) by a single account (MetroCard ID or OMNY ID), sorts them by time, and assigns each a "next trip" to be used later to infer the destination. Typically, the "next trip" is the next one in time. The last trip of a set gets matched to the first trip of the day so every trip gets a destination.

6. Origin-Destination Inference

The next step, origin-destination (OD) inference, uses the BIRDS (Bus Integrated Ridership Data Simulation) bus ridership model, which is the main model used for bus ridership analysis. This model was developed by NYCT Operations Planning (OP) and is now owned by the Data and Analytics group in Strategic Initiatives, housed in MTA Headquarters. It started being used in May 2022, and was based on an earlier AFC Bus Ridership Model built by OP in 2014. The BIRDS model improved upon the earlier model by using APC data to better estimate bus loads, fixing bugs with very high non-payment rates of over 50 percent, being much easier to maintain, running faster, and having slightly smarter OD inference.

Generate Dynamic Bus Transfer Links and Running Shortest Path Network

This step outputs the best guess of unlinked origin and destination stops for each leg of a passenger trip, walking links, and journey paths, which enable an individual passenger travel path to be determined. It uses the boarding stop chosen from candidates identified in Step 4. The process seeks to minimize the distance between the destination of a trip and the origin of the following trip. To do so, it uses the dynamic transfer link method to generate a partial transit network specific to the fare card itinerary and feeds it into Dijkstra's shortest-path algorithm to find the path with a shortest path assumption. This model assumes consistent passenger use of the same MetroCard or OMNY card, which includes a unique serial number in the transaction record, throughout the day.

When a trip has APC data, the process limits origin options to stops with APC boardings, and limits destination options to stops with APC alightings. When there is no APC data, and there are two options that are very close, the model chooses the one that is most common in aggregate

APC data. About six to seven percent of trips cannot be successfully assigned an origin and a destination, such as from single-ride tickets.

The dynamic transfer link method is a highly optimized network-generation tool that restricts shortest-path problem sizes when solving for the exact trajectory of a specific fare card given fuzzy information about boarding locations. Instead of creating walking links statically, the method takes advantage of the known fuzzy boarding locations of the next bus trip (due to a six-minute error margin in farebox timing) or subway station and creates walking links dynamically while computing the shortest path. It only generates possible walking links that meet predefined criteria (typically, distance between bus route and subway system or between two bus routes).

Maximum thresholds were set for time and distance to determine whether connections between different waypoints are valid. Walking distance thresholds were determined by conducting a sensitivity analysis on observed fare card trajectories from the percentage of identifiable passenger trips. If the distance between inferred alighting and known boarding stops is greater than the maximum transfer length, then an error condition is assumed, and the passenger's alighting stop must be determined without reference to the next trip. 70 percent of passenger trips were identified with a maximum transfer distance threshold of 0.25 miles for data from selected May 2014 weekdays, increasing to 79 percent for 1.9 miles. Since additional identified trips rapidly approached a point of diminishing returns when increasing the threshold to three miles, 1.9 miles was chosen as the transfer threshold. The transit routes in the model are obtained from GTFS data.

Bus-to-bus transfers

For bus-to-bus transfers, AFC data provides boarding time stamps to the nearest six minutes that allow BusTime to geocode to the nearest 0.5 mile (about three to five possible stops). The dynamic transfer link method then produces a limited custom network on the basis of the first bus trajectory (from the BusTime boarding location) and the few possible second bus pickup stops, solves for the shortest walking path on this small network using Dijkstra's algorithm, and yields the transfer point and the alighting point from the first bus.

Bus-to-subway transfers

For trips from bus to subway, the bus alighting stop is determined by solving for the shortest walk between route-specific possible alighting stops and the actual subway entrance logged by AFC, which provides definitive geographic locations. Similarly, a small network is generated with

the dynamic transfer link method, which is then solved for each transfer with Dijkstra's algorithm.

Subway-to-bus transfers

For trips from subway to bus, a daily lookup table is generated by finding the closest subway station within a given distance of each bus stop. Of the three to five possible bus boarding stops identified with BusTime data, the one closest to the subway is selected. An earlier method that used schedule-based shortest path to determine exit stations was scrapped because it required excessive computing time.

Other transit systems using MetroCard

Many Staten Island residents daily use the Staten Island Ferry to make transfers between the subway or buses in Lower Manhattan and buses or the Staten Island Railway (SIR) on Staten Island. A pseudo-St. George bus stop was created with South Ferry geocodes since the physical distance between South Ferry and St. George is greater than the 1.9 mile-transfer threshold. When dynamic transfer links are created from St. George to Lower Manhattan transit services, pseudo-locations are used to represent the Staten Island Ferry.

For the SIR, entry AFC transactions are not considered differently than that of a subway entry since most boardings from suburban stations have destinations of either St. George or Tompkinsville. Exit transactions are considered as boardings at any other non-fare control stations. For exit transactions, dynamic transfer links from buses include all possible transfers from any downstream bus stops within 1.9 miles of suburban SIR stations.

Before the BIRDS ridership started being used in 2022, the process accounted for previous swipes made on Port Authority Trans-Hudson (PATH), Nassau Inter-County Express (NICE), and Westchester Bee-Line. If a rider's previous swipe was at a PATH station, the rider was assigned to the bus stop closest to that station. The six PATH stations in Manhattan were replaced with the closest MTA subway station. All New Jersey PATH stations were replaced with the World Trade Center or 34th Street-Herald Square subway stations, which were likely Manhattan destinations of PATH passengers.

NICE bus did not provide similar automatic vehicle location data. For transfers to or from NICE, on the basis of the NICE route provided in AFC data, all stops on that route were considered possible boarding stops. When transfers were made between a NYCT or MTABC bus and a NICE

bus, dynamic transfer links were generated between each possible alighting stop on the NYCT or MTABC route within 1.9 miles of each possible entry stop on the NICE route. For transfers to or from Bee-Line Bus in Westchester County, the Bx16 bus, which operates along the Bronx–Westchester County line, was used as an approximate geographic representation of all Bee-Line routes.

Select Bus Service

For Select Bus Service (SBS), passengers who pay with MetroCards dip their cards into MetroCard fare machines on the sidewalk to pay their fares and must retain receipts for random on-board inspections. Because SBS buses no longer record passenger information, special AFC data processing is required. Because the sidewalk fare machine locations are static, specific boarding stops are determined. As such, for SBS boardings with MetroCard, the origin is known, but not the specific bus that was boarded.

If it is assumed that an SBS passenger always boards the next arriving bus, onboard passenger loads can be determined. Dynamic transfer links are created as for regular bus transfers. The program best guesses which bus was boarded based on the MetroCard six-minute window and bus arrivals.

When there is no APC data, the model makes a weighted random choice based on what percentage of the headway between bus arrivals is in the six-minute window. This assumes that the person could have randomly arrived at any time in that six-minute window and would take the next bus to arrive at the stop. If there is APC data, the program chooses the bus with the greatest difference between APC boardings and APC boardings already assigned to the bus at the stop. When there is a combination of buses with and without APC data in the six-minute window, the program does a random draw weighted by both headway and APC data.

Return to origin

The model assumes that the last trip of the day terminates where the first trip of the day originated if there is a second or more swipes in the same 24-hour period, and that midnight is the start of the day. If the last trip of the day is logical and not a possible return trip to the origin on that route, the rider's destination is assigned based on the proportion of people traveling that route. There are no corrections in the data for buses going across midnight.

One trip

Riders making only one trip for a day are considered as unknown riders and are assigned to a destination using a two-step probability function. The function considers the proportion of people who are known to get off the bus at the remaining bus stops and assigns riders to one of these stops.

Noncontinuous itineraries

In some cases, consecutive swipes or taps do not make logical sense, with consecutive swipes on bus routes miles apart being common in AFC data. These transfers are rare or geographically impossible. This can happen when passengers use another transport mode between consecutive swipes, changed MetroCards or OMNY mid-journey, when more than one pay-per-ride passenger shared a card, when a farebox failure resulted in a loss of transaction data, affecting the sequence of MetroCard swipes, or when AVL and GPS devices did not function as expected (travel paths cannot be built when bus trips do not provide passenger boarding locations and geographic fixes are not available). Generally, when a connection between consecutive swipes or taps is irrational, a passenger is treated as having paid cash or evaded the fare. If the boarding stop can be determined by AVL, then a boarding stop is assigned; otherwise, both boarding and alighting stops are factored from OD patterns generated from other identifiable trips.

7. Scaling for Fare Evasion and Missing Data

This step scales data on bus trips without APC to account for trips that either could not go through the OD inference process, such as trips where fares were not paid, or that were not assigned an OD during that process. In addition, for bus trips with APCs, these trips are assigned ODs through the creation of synthetic trips to match the APC on/offs at each stop for that trip.

Trips without APC

For bus trips without APC, data for fare payment on buses is used. However, passengers who board without paying using regular MetroCards or OMNY, such as people who pay with single-ride tickets, senior return tickets, or cash, or who evade the fare, do not generate transaction information, and therefore there is no OD information available for them. In addition, not all trips with OMNY or MetroCard are assigned an OD during the inference process.

Even though passengers who do not board buses by paying with MetroCard or OMNY might have different OD patterns, the differences are assumed to be insignificant because MetroCard

and OMNY account for most travel, and generated OD links match land use patterns well. As such, these passengers are assumed to have trip patterns similar to those of MetroCard and OMNY passengers.

The percentage of riders who pay in cash is obtained from bus farebox data from OMB that is provided on a one-month lag (it was provided on a multiple-month lag before 2019). OP determines the percentage of riders using cash for each route based on this data, dividing the number of cash riders by the sum of cash and MetroCard and OMNY riders.

The percentage of non-paying riders on each route has been determined from different sources over time:

> Originally, the scaling factor came mostly from ongoing OP ride checks for fare evasion, since APC coverage was sparse. OP had determined the percentage of non-paying riders on each route through these surveys. These riders included fare evaders, employees who did not swipe, and children under 44 inches tall who are not required to pay. OP calculated the percentage of the riding population these group represented for each route using an average of the prior three years of surveys. The fare evasion ratio by bus route was used to further adjust passenger trips. For routes with insufficient survey data, borough-wide survey-based evasion rates were used.
>
> Since OP stopped doing ride checks in March 2020, a combination of APC data and past ride checks started to be used. For routes with no or sparse APC data, current borough-wide APC fare evasion data was scaled by the ratio of pre-COVID route fare evasion to pre-COVID borough fare evasion.
>
> Beginning in July 2024, old ride check data stopped being used entirely, and APC-derived data started to be used for all routes. Since APCs are not installed or running properly on every bus, total ridership needs to be estimated by estimating the ridership per bus for trips without it. If there are enough trips with APCs, the average ridership per APC bus can be used as a stand-in for the ridership on non-APC buses.
>
> To get a sense of the error being introduced by using this estimate, each day, the buses are bootstrap resampled with replacement, with ridership per APC bus recalculated 1,000 times. Bootstrap resampling is an alternative approach to the standard inference

approach to estimate the nature of the sampling distribution for total ridership per bus. This approach provides a better representation of the variance of lower frequency data. The 5th and 95th percentiles of those samples are then used for the bounds. If the lower bound goes below the total number of riders counted on all APC buses, the lower bound is set to that number. In addition, if there is only one bus in the sample used for a given scale, its bounds are set to NULL.

Known ridership on trips with APC and the known number of trips on each route are used to estimate the total number of boardings on each route. A scaling up procedure is done at the route-direction-hour level. Then, this is aggregated to daily, weekly, month, and a month-weekday-hour level. At each level of aggregation, trips without APC are imputed by using the average riders per borough, with the bounds on that identified in the bootstrapping procedure. For routes with low APC coverage, ridership is imputed using the borough-service type or borough average, based on the amount of coverage in that source. The resulting dataset has a ridership estimate for each of the routes identified in bus matching, at each of the aforementioned time grains.

An important step in the procedure is reducing the total number of trips found in Bus Matching by making use of APC. Bus Matching occasionally finds trips that are entirely run with not-in-service head signs; however, some of these are actually in revenue service. To account for this, the proportion of not-in-service and low APC ridership (<2 riders) buses is identified for each route direction. That proportion is then used to reduce the total count across all (APC and non-APC) trips.

The final step merges the above data with the paid data sources at each of the relevant time grains. Since cash data is only available at the route level, consideration of route direction is dropped at this point. With paid and total boardings next to each other, an estimated fare evasion can be calculated. Since this involves identifying total trips run, this may slightly vary from the official fare evasion numbers.

Next, to account for fare evasion, cash passengers, and AFC trips that could not be assigned an OD during the inference process, boardings, alightings, and load are scaled up through the addition of trips synthesized according to MetroCard/OMNY OD patterns, with these trips randomly duplicating the trip pattern of a rider paying with MetroCard or OMNY.

Trips with APC

APC data only produces bus- and stop-level ridership and does not capture passenger ODs. As such, for trips with APC data, synthetic boardings and alightings are added to match APC on/off totals at each stop.

The process loops through stops with APC ons greater than assigned ons, creating synthetic trips originating at the stop one at a time until every stop has a sum of AFC boardings plus synthetic boardings equal to the APC boardings. The process chooses the destination for synthetic trips based on several criteria:

1. The stop's "distance" (in number of stops) from a destination of AFC passengers boarding at (or near) the synthetic trip's origin. The idea is that if this destination is a common inferred destination for paid passengers boarding at origin, it is likely a common destination for unpaid passengers boarding at the origin stop as well.

2. The difference between APC alightings at stop and AFC plus synthetic alightings at stop. That is, the "remaining" synthetic offs to be assigned there. This must be greater than zero for the stop to be considered.

3. The total number of alightings at the stop. Stops with more total alightings are more likely to be an unpaid rider's destination, all else equal.

A score is calculated for each possible destination for the synthetic trip based on these factors, and that with the highest score is chosen as the destination.

8. Averaging to create a representative day

Since there are relatively large daily variations in bus ridership by trip and OD pair, and there are variations due to trips the ridership model is unable to infer, which is particularly apparent for routes with few riders per OD pair, an average weekday ridership dataset is used to produce ridership metrics.

Using an average smooths out the effects of weather conditions, nonrecurring noise (e.g., street congestion), special events, and nonroutine customer travel. Averaging also helps fill in for missing data. On any given day, about ten percent of bus trips, tending to be randomly distributed throughout the fleet, are missing BusTime data, and averaging over multiple days

allows there to be reliable data for every bus trip. This also corrects for missing APC and AFC data.

This average weekday ridership dataset uses at least 20 days or a month's worth of Monday-to-Thursday ridership for each stop pairing on each scheduled trip. Fridays and holidays are not included because they often display different travel patterns than other weekdays. In some cases, days surrounding holidays are also excluded if it seems likely that the impact of the holiday will extend beyond the day itself.

Because of the computational resources necessary to process ridership data, the ridership used for daily calculation of metrics is typically on a one- to two-month lag, though this used to be done on a three- to six-month lag.

Before the use of the BIRDS model, since schedule-making and service planning primarily used data from high-ridership months only, the older model generated ridership data for a limited range of dates. These dates typically consisted of one to two months in the spring and fall, when passenger volumes are highest, and a similar timeframe in summer, which is subject to distinctly different ridership patterns with no school ridership, more leisure trips, and fewer people working at any one point. Data for July and August had been calculated using ridership from the prior summer. Although bus ridership in New York is fairly stable, especially at the granular level, there is some variation month-to-month which this process does not capture.

Since bus assignments differ each day, schedule run number and scheduled trip start time are used as keys to merge data from multiple days in a database. This ridership is then averaged, and a weight is provided, representing the average number of passengers for each OD and scheduled trip pair for a bus route. The weight is stored to four decimal places because many OD and trip combinations include significantly less than one passenger on average. All combinations are assigned a weight, though not every OD-trip combination exists within each individual day. This fact, in combination with the large number of bus stops in the system, means that the total number of records in the representative day dataset is almost five times larger than that of any single day.

The schedule used for the entire process is the base schedule for that pick, which does not include supplements or other modifications. As such, the schedule that actual data is based on may differ from the assumed schedule.

As an final step, before Passenger Wait Start Time Generation, for each route, a set of all schedules (defined as a route, pick, day type, and school open or school closed status combination) for the time period for the average day is generated and the statistical mode (the most common schedule type for the daytime and time period) is selected. These schedules are used to select the correct average day entries for that route, assuming that the most frequently occurring schedule will represent typical weekday service, even if it is represented by a school closed schedule.

9. Passenger Wait Start Time Generation

Before the metrics are calculated, there is an intermediate process for estimating when each volume of passengers assigned to a given bus trip and OD pair arrives at the bus stop to start waiting for the bus. The passenger count for each pair is split into individual passengers; the fractional remainder of the passenger count, if any, is treated as a single fractional passenger entry. This assignment process is necessary since fare collection for buses mostly occurs on board the bus, which therefore only produces observations of boarding times.

The model finds the first bus in the schedule that immediately precedes the passengers' scheduled boarding time and uses this to calculate their headway. Once a headway is established, the volume of passengers assigned to each pair are assigned a bus stop arrival time (wait start time). Wait start time is used to determine which bus a passenger would board given the actual service for a particular day: it dictates whether a passenger misses a bus if it arrives too early, or boards an earlier bus if it arrives sufficiently late.

In order to most accurately model human behavior, passengers within each scheduled headway are randomly assigned arrival times over a theoretical distribution selected based on the length of their particular headway. This method is designed to approximate the ways in which passengers respond to maximum possible wait time: it assumes that they arrive without consulting the schedule up to a certain headway length, after which progressively more passengers use the schedule to time their arrival to shorten their waiting time. The random arrival methodology applies an empirical model developed by Fan and Machemehl in the 2009 paper "Do Transit Users Just Wait for Buses or Wait with Strategies? Some Numerical Results that Transit Planners Should See" that lays out the shape of these distributions.

Passengers with headways under 11 minutes are assumed to arrive at random, and thus are uniformly distributed over their headway. If the headway is above 38 minutes, the headway is capped at 26.6 minutes and the passenger is randomly assigned a value within 26.6 minutes, as beyond that point passengers arrive non-randomly, being entirely coordinated with the schedule to avoid waiting an excessive amount of time. If the headway is above 11 minutes, the headway value is transformed based on the equation of (137/60+0.29*H)*2 where H is the headway, in minutes. This value represents the expected increase in the percentage of passengers coordinating their arrival. The passengers are then randomly assigned a value with this new transformed value.

This function, where H represents headway in minutes, describes the three separate cases:

$$Wait\ Time_{max} = 2\ \times \left(\frac{137}{60} + 0.29 \times H\right),\ when\ 11 \leq H \leq 38$$

$$Wait\ Time_{max}\ = H,\ when\ H\ <\ 11$$

$$Wait\ Time_{max} = 2\ \times\ 13.3,\ when\ H\ >\ 38$$

As described above, headways are capped at 38 minutes. Since the MTA has a policy by which local bus headways during the day cannot exceed 30 minutes, few local passenger trips, except those at the start of the service day, reach this threshold. On the other hand, express routes often have headways of 60 minutes during off-peak hours. These passengers typically arrive very well coordinated to the schedule, likely waiting far less than the 13.3 minutes on average that the headway calculation assumes. The current model likely underestimates ABST for these passengers, as they would likely not board a late bus which arrives at the beginning of the model's wait start time.

The methodology does not account for the change in rider behavior induced by real-time arrival information since current research does not provide sufficient consensus on how to quantitatively model arrival times when passengers are exposed to real-time information, which allows customers to coordinate the time they arrive at their stop with that of the bus, regardless of the schedule, effectively reducing their waiting time.

The algorithm's assumption that passengers arrive at random when headways are short and coordinate with the schedule when headways are long does not hold up when passengers arrive

in a coordinated fashion when buses are held to connect. The main example of this in New York City is the Staten Island Ferry, which operates between South Ferry in Manhattan and St. George Terminal on Staten Island. From there, a network of bus routes and the Staten Island Railway carry passengers across Staten Island. The ferry operates on a 15-minute peak and 30-minute off-peak headway, and both the buses and trains at St. George will wait for late ferry arrivals. These buses depart in one large "pulse" once all passengers have transferred from the boat. As detailed ferry arrival data was not available, St. George Terminal was treated the same as any other stop. Because late departures from the terminal often do not actually result in significant passenger ABST, the current model overestimates this metric for the routes involved.

10. Calculation of Metrics

To calculate the customer journey time metrics for a given day, each passenger from the representative day of trips is assigned both expected and actual wait and travel times. They are first assigned to scheduled buses before being assigned to actual buses. The scheduled times are then subtracted from the actual times to calculate how much more or less time each portion of each person's trip took. If a passenger's alighting time crosses midnight, 24 hours are added to their alighting time for the purposes of metric calculation. People assigned to trips in the reverse direction, typically on loop routes, are thrown out.

Assignment to Scheduled Buses

Passengers are first assigned to scheduled buses, making use of the Wait Start Time and the schedule. To determine the expected waiting time and trip length, passengers are assigned to the next bus scheduled to arrive after they reach the stop, of those buses that also serve their destination, except in situations described below.

If each passenger's origin and destination are in the schedule, every bus that serves these locations is iterated through until the first bus that is scheduled to arrive immediately prior to their assigned board time that fulfills the bus type requirement (detailed below) is assigned to that passenger. This process does not order buses so all buses must be checked. If no bus fulfills these requirements, the passenger is assigned a headway of 38 minutes.

When multiple buses that serve the passengers' destination arrive at the passengers' origin simultaneously, the passengers are assigned to one at random, regardless of route. This is meant to avoid loading all passengers onto the first ordered bus, therefore inaccurately representing behavior and negatively affecting capacity constraints. If no suitable bus is found,

the passenger is not assigned a bus and they are marked as unassigned (and do not go through the rest of the assignment process).

The process also checks whether the bus is in service. Buses that are deemed to be out of service at either the passengers' origin or destination are ignored, with passengers assigned to a subsequent bus. This can be caused by known missed trips from bus operator or equipment shortages, equipment failures, or buses turned around before completing their scheduled trips as a means of managing service.

Schedule data

Passengers' arrival times and travel behavior are identified using the regular weekday schedule in effect for the data source period but are matched to the daily operating schedule. This ensures that schedule changes are appropriately captured and that the metrics do not penalize for planned service changes.

MTA bus schedules do not usually change a lot day-to-day (outside of between picks), so it is assumed that within a pick, riders do not change their ridership origins and destinations significantly. Passengers are assigned on a given day to the buses that ran on that day, which may be different routes day-to-day.

Because these two schedules may not be the same, passengers may be assigned to a different route than originally intended in corridors served by multiple routes. In the case in which there are two or more buses with the same arrival time, the passenger is assigned to one at random.

Exclusions

In the event of a reduction in the hours of operation of service or stop omission whereby a passenger has no bus available to take, they are excluded from all further calculations.

Bus type

On corridors with express, limited-stop service or SBS operating in conjunction with local service, passengers assigned by the ridership model to limited or SBS trips are presumed to not board local buses. Passengers who are assigned to a local bus or school service bus in the ridership data may board any type of bus except express buses (local, limited, SBS) if one of those buses is scheduled to arrive first and their destination is

served by that bus. Before January 1, 2019, these riders could board express buses in the model, and as of that date, all passengers were able to board School Limited trips in the model.

This is intended to account for the assumption that riders assigned to faster service are less likely to accept a slower option, whereas local service riders would accept faster service. However, this also implies that these passengers will never board a local bus if there is a disruption on the limited or SBS service or if real-time information indicates that the limited or SBS service will arrive significantly later.

Assignment to Actual Buses

Passengers are then assigned to actual buses**,** independently of their scheduled bus assignment, using a similar process. For each passenger, a list of buses that serve their origin which arrive at their stop is provided, ordered by their arrival time. The script searches the list of buses, starting with the soonest to arrive immediately before the passenger gets to the stop. People will only get on a bus if it serves both their origin and destination. The bus must allow people to board at their origin and alight at their destination to avoid impossible trips (e.g. passengers are blocked from boarding a bus at an alight-only stop). If no bus is found that serves their origin, the passenger is not assigned and is marked as such and does not go through the rest of the assignment process.

If no suitable buses are found in that entry, the script iterates to the next-soonest arriving one. In service status is not checked to account for the fact that a bus two headways in advance is less likely to overtake two buses, regardless of the status of the bus in front of it. If at least one of the buses fulfills all requirements for boarding, including being of the right bus type, all of the entries that are suitable are stored. When multiple buses that serve the passengers' destination arrive at the passengers' origin simultaneously, they are assigned to one at random, regardless of route.

The script then checks for the capacity of the bus, defined by a table with bus capacities based on where it is a standard, articulated, or express bus. The vehicle type is based on the size of buses assigned to the bus route, as defined in a table, not on the actual bus number. Capacity limits were defined as 100 for a standard bus, 160 for an articulated bus, and 75 for an express bus between August 1, 2017 and June 30, 2019. On July 1, 2019, the limits were changed to 85,

130, and 67, respectively. This process is conducted in stop-arrival order, such that buses fill up as they move along the route. If the number of passengers on the bus exceeds full capacity, the passenger is assumed to be blocked from boarding and must wait for the next bus that is not at full capacity.

Exclusions

If the next bus that arrives at a passenger's stop after they have started waiting is indicated to be out of service at either their origin or destination, they wait for the next bus. The destination requirement accounts for the fact that it cannot be assumed how a rider completes their trip, but results in the appropriate wait time penalization for being kicked off a bus.

Buses without observations for the bus, though they are expected to be in service, along with passengers for whom the next bus has these values, are excluded from metric calculation. In addition, passengers on trips are thrown out if the immediately preceding bus that arrived before they arrived at their stop had these values since passengers could have boarded that bus if it ran sufficiently late. If multiple buses arrived at the same time as the previous bus and any of them had no observations, the passenger is thrown out. This step is done since it cannot be known whether a passenger boarded these buses, and, if they did, there is no data that allows their CJTP values to be calculated.

Trips can be thrown out if the bus is matched in the wrong direction, with a passenger's trip matched so they arrive at the destination prior to departing from the origin. This only refers to the specific OD pair and does not drop the trips that may match out of order for intermediate stops.

Bus stop changes

The model uses the boarding and alighting stops of individual passengers in ridership data to perform assignment. However, due to the availability of ridership data, the schedule and actual buses being assigned to often do not match the schedule for which the ridership data is generated. Stop IDs change, sometimes significantly between picks, and large numbers of passengers could be mismatched. To prevent this, passengers

whose stops were removed are matched to the geographically closest stop, up to 1,000 feet.

Generally, this process is run whenever a new schedule is loaded and the ridership data remains the same, or vice versa. Due to known issues in the model, translation rows may be produced if the ridership data is run against the same pick it came from. There is no point in doing this as those passengers were likely assigned the wrong stop in the model to begin with.

Ridership information is by route-OD, which means that all service changes need to be accounted for. The MTA routinely makes minor stop changes to resolve localized congestion, account for long-term construction, and address other operational issues. These changes usually involve the relocation of a stop across the street or down the block. Routes may also be renamed.

For these minor changes, a process was developed that compared the stops listed in the ridership model with the stops served in the new schedule each quarter. Passengers whose origin or destination no longer exists on their route are reassigned to board or alight at the nearest stop served by the same or renamed route in the same direction, computed using the Haversine formula.

A cap on walking distance can be specified. For New York City, 1,000 feet was chosen given that passengers are likely to consider other transit options in the dense New York City network if it becomes necessary to walk further.

Calculating metrics

The results of these assignments are stored in a database on a per-passenger (whole or fractional) basis.

ABST and ATT are calculated as the difference between the scheduled and actual wait and trip times respectively. Aggregate ABST and ATT are weighted averages of these values, based on the average actual number of passengers assigned to each trip. When aggregating, both ABST and ATT are capped at 90 minutes. Beyond this point, it is presumed that the passenger likely sought alternative means of transportation and did not wait for the next bus. In practice, such cases are relatively uncommon.

Passengers who could not be assigned an actual or scheduled bus are excluded from calculation; this could result from lack of service to a particular stop on a day, such as school stops on non-school days, from the last trip of the day running early, from a reduction in the scheduled hours of operation of service, or if a stop pair is no longer in the schedule. In some cases, this process may underrepresent the negative impact of the disappearance of a final trip, but this occurrence is very uncommon and assigning an ABST value to these passengers both requires a strong assumption about the magnitude of their wait and does not have a significant impact on the value of the aggregate metric.

Passengers who have negative ABST as a result of boarding earlier buses that are running late bring down the overall ABST values. ABST reflects both how many passengers are affected and how significantly. The passenger-weighted nature of ATT means that it is more sensitive to customer experience than metrics such as bus speeds; if a relatively short segment with high ridership sees a significant impact on its speed, the overall speed value may show only minor changes; by contrast, ATT would capture how many passengers were negatively affected.

Statistical and Analytic Issues

The data is available starting in August 2017. Data in March 2020 is calculated only for March 1 to March 13 since the metrics depend on ridership models to calculate customer travel times to compare actual service to scheduled service, and those models could not properly account for the significant changes in ridership levels and travel patterns that began mid-month with the outbreak of the COVID-19 pandemic. Data is not available from April 2020 to August 2020 as a result of the suspension of fare payment due to COVID-19, which meant the model could not run.

The data is broken down at the monthly level, and there are a few other factors to consider when working with the data:

- The dataset combines some routes together as they have few daily trips, leading to sample sizes that are too small to be displayed individually.
- Some routes do not have data available because they have atypical characteristics that make processing performance data inconsistent.

Each route is assigned to a single borough based on the letters used for the route number. For example, data for the M60-SBS is included in the total when Manhattan is selected, but not if only Queens is

selected. Similarly, the Q54 is included in Queens, and the S79-SBS is included in Staten Island. Express routes are counted in their outer borough, so the BxM7 is a Bronx route and the X27 is a Brooklyn route.

ABST, ATT, and CJTP are measured for passengers whose wait start times are between 4 a.m. and 11 p.m., with the peak period defined as 7 a.m. to 9 a.m. and 4 p.m. to 7 p.m.

On August 1, 2025, bus performance indicators from January 1, 2019 to present on NY Open Data were updated. The updated numbers come from a new bus performance indicator calculation process that, through the use of improved algorithms and enhanced computational capabilities, produces results that more accurately reflect the customer experience. Among these process improvements, those with the largest impact on the metrics were:

- More accurate identification of bus trips from GPS ping data, which in turn permitted better identification of scheduled bus service not operated.
- Better estimation of the time buses depart the first stop of a trip, and the time buses arrive at the last stop of a trip. This made possible more accurate estimates of customer journey times.

These new numbers are shifted from the old numbers, but should follow the same key trends 2019-present. By the end of 2025, the MTA intends to extend this metric recalculation to the years 2015-2018; in the meantime, metrics from before 2019-01-01 should not be directly compared to those after, as they represent the results of different methodologies.

In the summer of 2025, the Queens Bus Network Redesign (QBNR) took place to bring Queens faster, more reliable service with better connections. Changes were phased in by route on June 29 and August 31, which resulted in changed, discontinued, and entirely new routes. Some routes have no changes and are comparable from before and after this time period (ex. Q70); some are mostly the same but rerouted in small sections (ex. Q23); and one reuses a previous route name but is not the same route at all (Q48). New Rush routes were introduced, and there were also changes to bus stops, including stops being relocated or removed. Caution should be exercised when comparing route performance before and after the redesign took effect. More details about routes changes can be found on MTA's website on the Queens Bus Network Redesign Service Changes page.

Limitations of Data Use
There are no limitations on the data at this time.

Release Notes

Version 1.0.0 release notes instituted and methodology expanded (09/04/2024)

Version 1.1.0 documentation updated and 2019-present data rebuilt (8/01/2025)

Version 1.1.1 documentation updated for QBNR (8/19/2025)

Version 1.1.2 bug fix to address SBS school trips incorrectly grouped under LCL/LTD; combine temporal bounded versions of data (12/16/2025)