SUMO User Conference 2025
Conference paper
https://doi.org/10.52825/scp.v6i.2613

Published: 15 Jul. 2025

Modeling Bus Traffic for the Berlin SUMO Traffic Scenario

Moritz Schweppenhäuser[1,*], Timo Großmann[2], Karl Schrab[1], Robert Protzmann[1], and Ilja Radusch[3]

[1]Fraunhofer FOKUS, Berlin, Germany

[2]Technische Universität Berlin, Germany

[3]Daimler Center for Automotive IT Innovations (DCAITI), TU Berlin, Germany

*Correspondence: Moritz Schweppenhäuser, moritz.schweppenhaeuser@fokus.fraunhofer.de

Abstract. The digital transformation of the transport sector in our cities will be led by the deployment of large-scale digital twins, interacting with their real world counterpart to model, predict, and improve movements and reoccurring patterns. Traffic simulation is an essential tool in this area. While both macroscopic and microscopic simulations are possible, only the latter provide enough detail to realize sophisticated Intelligent Traffic Systems (ITS). One of the biggest challenges is accurately modeling road traffic on a large scale due to limitations in both reliable data sources, as well as the quickly increasing complexity of size. Only a handful of city-scale traffic scenarios exist, and only a few of them include public transport modalities. With this paper, our aim is to extend this list by integrating bus traffic within the Berlin SUMO Traffic scenario (BeST). We provide an overview of potential data sources and a detailed description of the applied methodology. As the scenario was initially calibrated with only individual private traffic, we conduct an evaluation on how the added traffic volume affects the stability of the scenario.

Keywords: Bus Traffic Simulation, Scenario Modeling, Eclipse MOSAIC

1. Introduction

Transforming urban transport systems into a more sustainable future requires forward thinking solutions, including electrification and concepts that improve the efficiency of traffic. Using simulation allows the implementation of such ideas while avoiding the large costs of performing field tests. These simulations need to take into account not only the traffic, but also require consideration of communication (e.g., via ITS-G5 or 6G), on-board or cloud applications, as well as electric mobility, and more. In an effort to combine best-in-place simulators from these different domains, we developed Eclipse MOSAIC [1] as a multi-domain, multi-scale co-simulation framework. One key aspect of MOSAIC, is its application simulator, which can deploy any user-defined application onto every entity in a simulation. These applications utilize a well-defined API to change routes, communicate with other applications, perceive surrounding entities [2] and much more. These strengths have been leveraged in the past for the implementation and evaluation of highway traffic control algorithms, ride-hailing strategies, or novel Traffic State Estimation (TSE) techniques [3]. The studies have been carried out by coupling MOSAIC's bundled application and cellular simulators with the traffic simulator Eclipse SUMO [4]. SUMO stood test of time and has established itself as the leading microscopic traffic simulator in the open-source domain.

Regardless of the researched solution, a calibrated traffic scenario is required as a baseline to draw significant conclusions. For the purposes of this paper, public transportation, more specifically buses, plays an important role, since we are interested in possible utilization strategies of the data generated by the public transportation mileage.

Currently, there are a handful of validated SUMO traffic scenarios based on real-world data, with different goals in mind. These scenarios also range in size, detail, and covered modalities. The Luxemburg SUMO Traffic Scenario (LuST) [5] and the Monaco SUMO Traffic Scenario (MoST) [6] were built with the intention of researching VANET applications. Another example is the Ingolstadt Traffic Scenario (InTAS)[7], which was built with the intention of evaluating novel traffic signal paradigms. The aforementioned scenarios were built with bus traffic in mind and, therefore, additional infrastructure (i.e., bus stops and bus lanes) was included in the network creation process. The actual traffic is then typically generated using some kind of demand data (e.g., census data, living areas, industrial areas, offices, etc.) to generate trips. The routes for those trips will then be calculated and optimized using dynamic user assignment to achieve a user equilibrium [8]. Similarly, the Berlin SUMO Traffic (BeST) scenario [9] was created. Compared to the other scenarios cited, the BeST scenario is much larger in both area and traffic volume, which is one of the reasons why at its inception no public transport was considered. However, due to recent research interests (e.g., TSE), an extension of the original BeST scenario with realistic bus traffic was required, with the aims of having accurate routes and schedules of buses and the buses having a minor impact on the background traffic. Unfortunately, simply regenerating the scenario is not an option, as significant manual labor was put into tweaking the network, as underlying data sources can be erroneous and lack detail. Hence, we are faced with the challenge of adjusting both the network and injecting additional traffic into an existing scenario.

To explain our procedure, the paper is structured as follows. In Section 2, we explain how the simulation of bus traffic differs from regular vehicles. Furthermore, we illustrate potential data sources and their conversion to formats readable by SUMO. Next, in Section 3, concrete approaches and challenges during conversion are explained. Afterwards, we conducted an evaluation of the resulting scenario and investigated both impact and timeliness of integrated bus traffic in Section 4. Finally, we draw conclusions and give an outlook on further advancements in Section 5.

2. Bus Traffic Simulation

The microscopic simulation of bus traffic differs in a couple of modeling aspects compared to regular car traffic. In terms of the car-following and lane-change model, buses can be treated as large and heavy vehicles, implying lower acceleration and deceleration, but otherwise being handled similar to passenger cars. However, buses require additional information to be modeled correctly. This includes additional infrastructure, such as designated bus lanes and stops, as well as routes and schedules. Bus routes (including information on where to stop) and schedules can generally be defined using regular vehicle semantics. For SUMO these are defined in a route file, of which a simulation can handle an arbitrary amount at the same time. In addition, SUMO defines bus stops in an additional file, where the IDs and lengths of the stops are matched to a lane and offset position on top of the road network. Consequently, when faced with the challenge of extending an existing scenario, the addition of bus routes, schedules, and stops is challenging but can often be handled programmatically, as they can be defined independently of existing configurations. In contrast, bus lanes are part of the road network (since they interact with other streets) and therefore need direct integration with the original network file, which proves to be problematic at times.

In Figure 1, we outline our planned procedure for integrating bus traffic within the BeST scenario.

- We describe the data sources used for the **Requirements** in Section 2.1.
- Applied **Conversion** and Merging strategies are described throughout Section 3.
- Finally, we test the resulting **BeST-Bus Scenario** in Section 4.

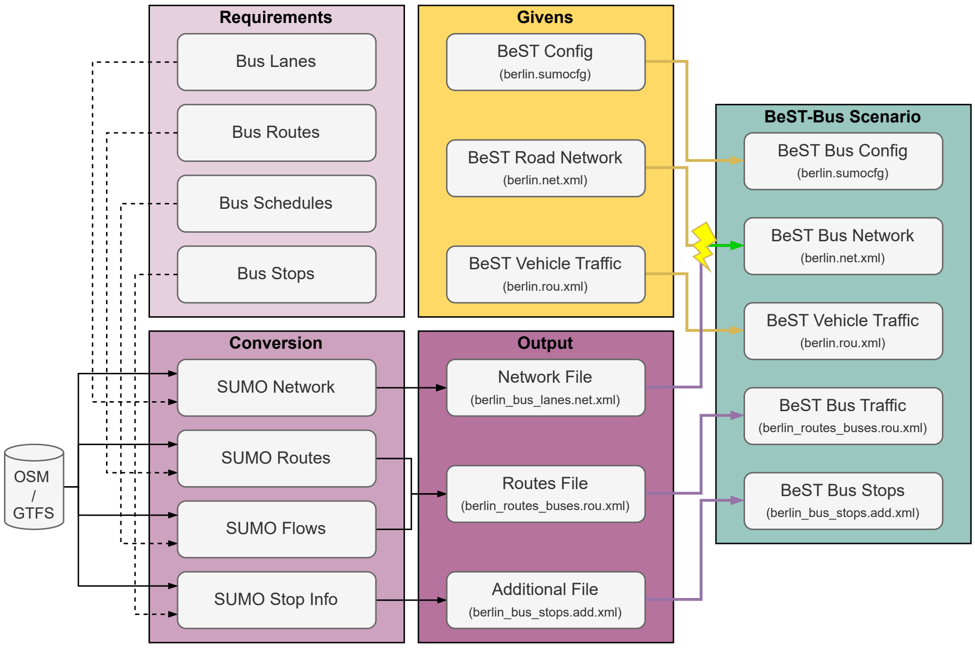


Figure 1. *Workflow for the transformation of the BeST scenario.*

2.1 Data sources

Accurate modeling of bus traffic requires reliable data sources. In terms of bus routes, stops, and schedules, this involves extracting bus routes for each line and direction, identifying the corresponding stops along with their geographical locations and timetable data. Regarding designated bus lanes, one needs to extract not only their geographical locations but also turning rules at intersections.

In this work, we considered *OpenStreetMap* (OSM) [10] and *General Transit Feed Specification* (GTFS) [11] data as potential sources. In addition, the Berlin Road Traffic Authority publishes an open dataset containing designated cycling and bus infrastructure (`https://daten.berlin.de/datensaetze/radverkehrsanlagen-wfs-0c37901e`). However, we abstained from using this dataset because it seems to be mostly integrated with OSM and additional conversion efforts would have been required.

OSM is the largest and best maintained global crowd-sourced geographic database. It provides a comprehensive and detailed representation of the Berlin road network, including the location of bus stops and the route information of various bus lines. Unfortunately, OSM lacks scheduling information, necessitating the use of a secondary data source to extract and merge transit schedules with OSM's bus routes and stop sequences. This limitation introduces the challenge of aligning OSM's bus stop sequences with actual scheduling data, requiring considerable effort and increasing the complexity of the implementation. Nonetheless, for the extraction of bus lanes, OSM proves to be of significance.

GTFS is a standardized format for public transit schedules and their geographical locations. Datasets can generally be categorized into GTFS Schedules and GTFS Realtime. While the latter pertains to up-to-date information to inform passengers in real-time of changes to the schedule, GTFS Schedule refers to static feeds, which include preplanned public transport schedules, fare information, and spatial details about the routes and stops. Each feed is provided as a compressed zip file that contains multiple tables stored in separate, simple text files. Similarly to relational databases, each table includes a key column that allows one to link information described in one table to the data described in another. The Verkehrsverbund Berlin-Brandenburg (VBB), publishes an open-source, weekly updated dataset (https://www.vbb.de/vbb-services/api-open-data/datensaetze/) in the GTFS format that includes information for all public transport modes in Berlin in Brandenburg.

2.2 Conversion to SUMO formats

To integrate the extracted bus traffic data, it must be converted into a format compatible with SUMO. The input files are defined using an XML-based format. As shown in Figure 1, a typical simulation scenario consists of a network file representing the underlying road infrastructure and a separate file specifying vehicles and their respective routes through the network. For public transport, an additional file is required to define bus stops, including their precise locations within the network. For the sake of enhanced readability and simpler corrections, we aim at a flow-based approach of schedule definition. Flows are defined with a designated start and end time, as well as a period.

SUMO already provides a wide range of conversion tools for the extraction of public transport schedules and route. Most prominently, **Gtfs2pt**[1] and **Ptlines2flows**[2]. Furthermore, **Netconvert**[3] is relevant when extracting topological information, such as bus lanes and stop positions.

Gtfs2pt, as the name suggests, aims to transfer GTFS data into SUMO readable configurations. The script requires a network, a GTFS dataset, and a day to generate 24 h of bus traffic in the form of a route and an additional file. The route file will include a separate vehicle definition for each bus, making the result for large scenarios cluttered. Per default, **Gtfs2pt** uses shortest path routing between bus stops instead of actual routes, which slows conversion and can lead to differing routes. Alternatively, it is also possible to provide a SUMO ptLines file, generated using **Netconvert** from OSM data. However, for this to work, the SUMO network must mostly match the OSM network (i.e., SUMO edge IDs must correlate to OSM way IDs). With the BeST scenario, this approach is not applicable because the network is partly outdated and adjusted from the original OSM data fetched in November 2020. **Gtfs2pt** disregards GTFS' shape

[1] https://sumo.dlr.de/docs/Tools/Import/GTFS.html
[2] https://sumo.dlr.de/docs/Tutorials/PT_from_OpenStreetMap.html
[3] https://sumo.dlr.de/docs/netconvert.html

information (traces of bus routes), which could avoid discrepancies between GTFS and OSM but requires potentially difficult map-matching.

Ptlines2flows is solely based on data extracted from OSM. In addition to the network and the aforementioned ptLines file, the script requires an additional file, which can be generated in the same **Netconvert** call. This script will generate flows for each direction of each bus line with a constant frequency. However, it does not adhere to actual frequencies and disregards differing schedules throughout the day. Furthermore, the same concerns of **Gtfs2pt** regarding the networks persist.

The restrictions and complexities of **Gtfs2pt** as well as **Ptlines2flows'** missing level of detail made SUMO's bundled solutions insufficient for the task at hand. Thus, an adapted approach is required that extracts the information and transforms the necessary data for all valid bus routes on the existing network.

3. Methodology

This section will explain our adopted approaches to integrate bus traffic into the existing BeST scenario. Referring to Figure 1, we opted to extract bus routes, stops, and schedules using GTFS data. These extraction steps are described in Section 3.1. Our procedure for integrating bus lanes from OSM is described in Section 3.2.

Throughout the work of the paper, we used SUMO version 1.22.0 and deployed the latest version of the BeST scenario. We are using GTFS data from the 07.12.2023 from VBB, as it is the day with the most bus activity. As the GTFS data set holds a large number of columns from all public transport agencies and modalities in Berlin and Brandenburg, we preprocess the GTFS data. Resulting in a filtered data set that includes 200 bus lines with 386 independent bus routes, 27,968 trips, 6389 stops, and 1479 shapes over a 24-hour period. Here, a bus line refers to a given number (e.g., 260, M45, N8, ...) of which several routes can exist. The numbers indicate that not all bus lines operate in both directions or travel on a circular route.

Where ever possible, we tried to adhere to the following *quality criteria* with decreasing priority:

- **Correctness**: The extracted bus traffic shall reflect the real-world routes and schedules as close as possible. This can be hindered by insufficient data sources.
- **Non-Intrusiveness**: The existing scenario shall not be affected significantly. This includes avoidance of network adjustments that invalidates existing routes and additional traffic, that causes major bottlenecks.
- **Traceability**: Throughout the conversion process we want to document occurring errors and misalignment, both textually and visually. This also means that generated output shall be formatted in a human-readable way.
- **Performance**: Deployed conversion processes aim for low execution times even for large-scale public transport volumes.

3.1 Extracting Bus Traffic

As laid out in the previous section, existing SUMO conversion tools do not meet our quality criteria sufficiently. Specifically, using shortest path routing between bus stops does not meet our *Correctness* quality criteria, and the application of OSM routes has been shown to be unreliable in conjunction with the BeST network. Hence, we decided for a map-matching approach using GPS traces provided within the GTFS data

Table 1. *Overview bus stop mapping and covered main shapes.*

	Number
Total Bus Stops	6384
Mappable Stops	6246 (97.8 %)
Bus Stops Outside of Bounding Box	4 (<0.01 %)
Non-Mappable Stops	134 (2 %)
Total Main Shapes	386
Main Shapes affected by missing Stops	29 (7.5 %)
Cut Main Shapes	19 (5.0 %)
Main Shapes Candidates	376 (97.4 %)

set. In GTFS, a bus route can have multiple attached GPS traces, which consist of a sequence of geographical coordinates. Therefore, per bus route, we select one GPS trace ($\hat{=}$ *main shape*) based on the number of attached trips (**Gtfs2pt** applies a similar approach), resulting in the 386 potential routes.

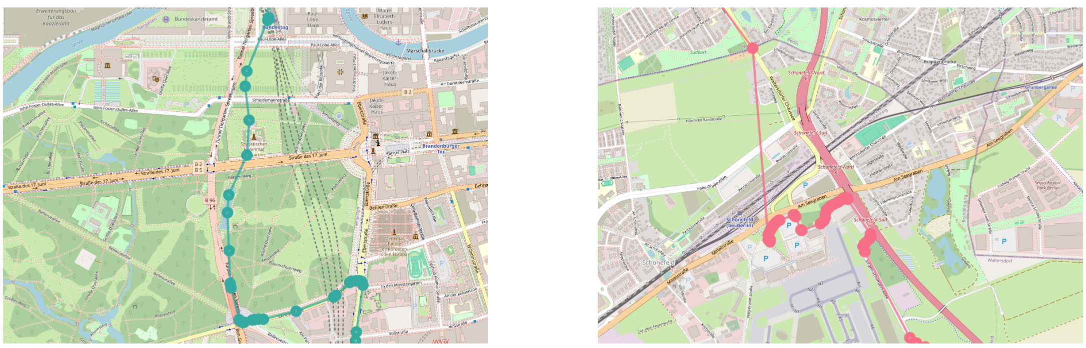
3.1.1 Placing Bus Stops

From the reduced shape set, we first apply a simple radius-based approach based on SUMO methods to try to map serviced stops to a nearby edge in the network. In case multiple stops inside a shape cannot be mapped to an edge, it can be assumed that the network is missing parts of the real road network, leading us to sift out respective shapes early.

Table 1 gives an overview on the number of stops that can be mapped to the network. Although network adjustments were made to facilitate missing stops, in the later steps, around 5 % of the bus routes are not considered.

3.1.2 Map-Matching Shapes

The success in transforming these trips depends on accurately mapping the shape onto the existing network. One issue is that GPS-based data can be prone to inaccuracies, which can affect the precision of the extracted routes. As illustrated in Figure 2, two examples highlight these challenges. The first involves a tunnel where the GPS signal deteriorates, while the second concerns a segment of a route originating from Berlin Airport that lacks GPS coordinates.

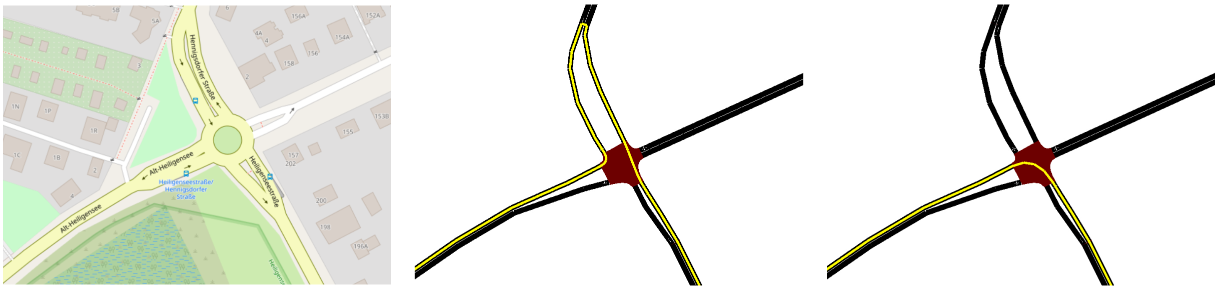


(a) *Tiergartentunnel* ***(b)*** *Route from BER airport*

Figure 2. *Inaccuracies of GPS coordinates in VBB GTFS data set (likely due to GPS inaccuracies).*

With this in mind, we try to apply map-matching for the remaining shapes to generate SUMO readable routes. We utilize the SUMO-native tools **Tracemapper** and **Routechecker** to first match and later validate routes. Initially, 244 main shapes can be successfully transformed into SUMO routes, around 63 % of the original routes.

Based on a manual review, two distinct error patterns in **Tracemapper**-results have been identified. The first pattern arises due to simplifications during the network creation process or due to slightly offset GPS coordinates. For example, shape 11967 (bus line N22) cannot be mapped, because of a simplified roundabout in Heiligensee (see Figures 3a and 3b). This type of error can be resolved algorithmically using junctions as the common denominator by cutting routes so that valid turning behavior remains, as shown in Figure 3c. As a result, 24 additional shapes were successfully salvaged, bringing the total to 268 (69 %).



(a) *Roundabout in OSM* ***(b)*** *Mismatched route at junction* ***(c)*** *Route after correction*

Figure 3. *First error pattern involving a simplified roundabout in Heiligensee and shape 11967 (line N22).*

The second pattern of errors occurs at the start and end of bus routes. Occasionally, after a trip has been completed, bus drivers will have a break in a designated area outside of the public road network, which often is included within GTFS traces but not covered in the network. Exemplary, Figure 4a illustrates this for shape 10868 (line 260) in Berlin Adlershof, the depicted u-turn is forbidden. U-turns were methodically omitted in the creation of the BeST scenario as it allowed for better results during the dynamic user assignment.

(a) *Shape 10868 in OSM* ***(b)*** *Found SUMO route for shape 10868*

Figure 4. *Second error pattern at a removed roundabout in Adlershof and shape 11967 (line 260).*

To resolve these errors, the unreachable segments of the route are trimmed and affected bus stops are relocated to the first reachable edge of the remaining route. With these adjustments, 89 routes additional routes are recovered, further increasing the number of valid routes to 359 (93 %).

For each of these routes, we map its GTFS stop sequence to the closest edges, ignoring all edges that do not belong to the route. Again, we utilize SUMO tools but increase the search distance to 200 m to accommodate potentially missing edges in the network. Through this step, 15 routes are excluded. For these routes, most often the bus stop locations are incorrectly placed with respect to the shape. Exemplary, we encountered a route in which one of the bus stops is placed according to a diversion plan, but the shape follows the plan without any diversions.

Finally, leaving us with 344 (89 %) successfully converted bus routes out of the original 386 main shapes. Note that initially this number was much lower and was only achievable through manual inspections of bus routes and the network to identify missing streets, service areas, and other problematic segments.

3.1.3 Bus Schedules

Finally, the schedule, including the earliest and last departures, as well as the frequency, is extracted from the GTFS data set, using the median intervals observed in the data set. Furthermore, flows are configured with departure times for each stop and a maximum wait duration of 20 s should a bus arrive after its' desired departure time. The resulting SUMO routes and flows are written to a SUMO route file (`berlin_routes_buses.rou.xml`), and the stops to an additional file (`berlin_bus_stops.add.xml`).

3.2 Network Adjustments

As the BeST network was generated without bus traffic in mind, it is missing crucial parts required for a accurate simulation. This includes designated bus lanes, private streets used by bus operators, as well as streets classified as *undefined* in OSM. This section explains our efforts at integrating these network elements into the BeST network.

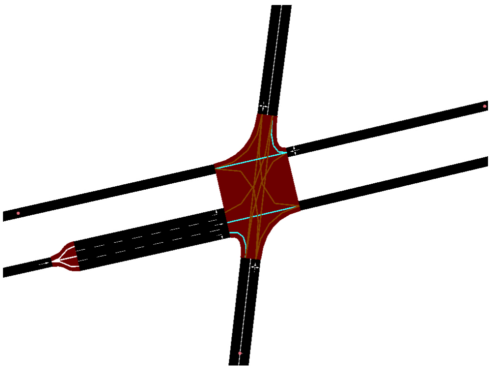
3.2.1 Bus Lane Integration

A realistic simulation of bus traffic involves the inclusion of bus lanes. As touched on in Section 2.1, using OSM is the most promising source for placement of designated lanes, as many edge IDs between the networks will match even though the BeST network has many optimizations. OSM uses three tagging scheme to represent bus lanes: the `"busway="` scheme, the `"lanes="` scheme, and the `"bus=*"` scheme. We use the Overpass API[4] to query the OSM database for these schemes. For Berlin, 1334 ways that include bus lanes are identified.

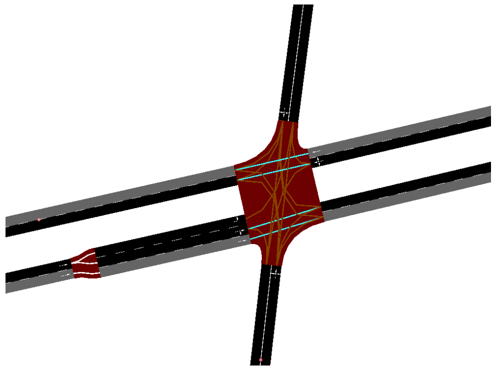
Converting these ways to a SUMO network implies using the native tool **Netconvert**, which results in the `berlin_bus_lanes.net.xml` network file. At the time of writing, not all tagging schemes are converted correctly. Though challenging, identifying and converting bus lanes, can be achieved through programmatic approaches. However, merging the existing `berlin.net.xml` with the created `berlin_bus_lanes.net.xml` is challenging. **Netconvert** can merge multiple networks and allows one to fuse overlapping elements with the commands `--junctions.join-same` and `--edges.join`. Unfortunately, in many cases, these options failed to properly merge the two networks, often due to minor offsets in coordinates, resulting in overlapping network elements.

[4] https://wiki.openstreetmap.org/wiki/Overpass_API

Even if we would have managed to identify all edges where additional bus lanes are present, we would have been required to adjust turning rules and traffic signals at junctions, which are often manually adjusted. In addition, partly due to the BeST network, partly due to inaccuracies in OSM, and partly due to conversion tactics of **Netconvert**, the number of lanes and designated bus lanes can differ between real roads and the BeST network (validated by satellite images). Figure 5 illustrates the complexity of the task.



(a) *Original BeST network*

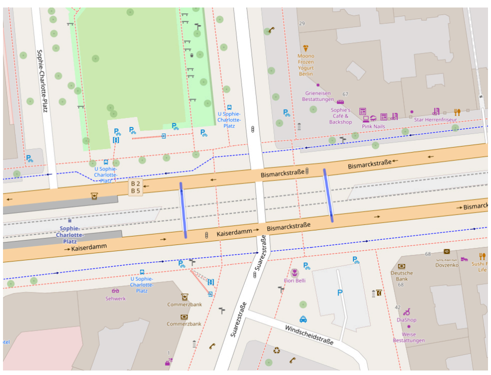


(b) *Updated BeST network with bus lanes*

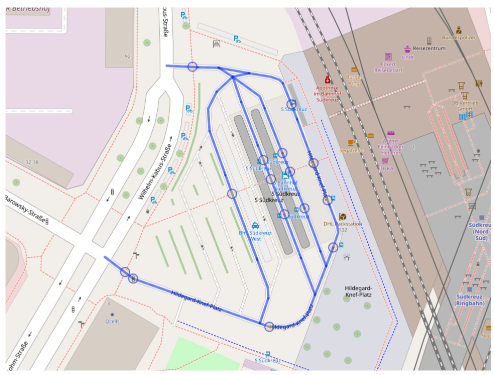
Figure 5. *Comparison of Kurfürstdendamm/Knesebeckstraße before and after addition of bus lanes.*

Due to the complexities shown, we were unable to find a fully programmatic solution for the integration of bus lanes. Instead, our integration efforts were focused on a single district of Berlin, namely Charlottenburg. We extract lane position from OSM as explained earlier, and manually review connections at junctions, additionally consulting satellite data. Eventually, 162 bus lanes were added to the network.

In addition to bus lanes, buses often traverse private road infrastructure marked as `service_roads` in OSM. At times, these small connecting streets (see Figure 6a) and private roads, particularly near railway stations (see Figure 6b), can be important. Similarly to bus lanes, we can use the Overpass API to identify all service roads that are part of a bus route. However, again, integration and adjustments of turning rules had to be done manually and were only introduced in Charlottenburg.



(a) *Small connectors on Bismarckstraße*



(b) *Rail station S-Südkreuz*

Figure 6. *Service roads used by buses (marked in blue) initially not included in the BeST network.*

Table 2. *Simulation setup.*

	BeST	BeST Bus
Total Number of Vehicles	2,246,926	2,272,641
Number of Cars	2,246,926	2,246,917
Number of Bus Trips	-	25,724
Number of Bus Routes	-	344
Number of Stops	-	6246
Time-to-Teleport	120 s	120 s
Simulation Period	24 h	24 h
Parking at Bus Stops	-	True
Simulation Duration	7 h	7 h

3.2.2 Manual Network Adjustments

Throughout the transformation process, key differences between the BeST network and OSM data (i.e., real world) became apparent, both due to the age of the network as well as assumptions made during its creation. To further increase the number of integrable bus routes, the network is updated at crucial points while ensuring that network modifications do not negatively affect the original vehicle traffic. An illustrative example is the Hertzallee, near Zoologischer Garten, which is a one-way street that allows buses to travel in both directions. Adding the opposing direction enabled 20 additional bus routes, as the Hertzallee serves as an important bus hub. Furthermore, we noticed that OSM ways tagged as *undefined* were initially not included but most often represent residential roads. By manually including many of these ways, we incrementally increased the amount of bus routes to the result of 344. Sporadically, bus traffic can significantly disrupt overall traffic flow, negatively affecting the entire scenario. To address these issues, the network was adjusted in areas where severe traffic jams occurred. One key adjustment involves converting yielding connections to non-yielding.

4. Evaluation

In total, we were able to incorporated 344 out of 386 bus lines, and 6246 stops into the BeST scenario, resulting in 24,780 additional trips. To assess the impact of additional bus traffic, a simulative evaluation is conducted. Furthermore, we want to assess how well buses adhere to their schedules. For comparison purposes, the original BeST scenario is used. Throughout this section, we will refer to the original scenario as "BeST", while the edited version will be called "BeST Bus".

4.1 Evaluation Setup

Simulations are setup with a step-size of 1 s, and a time-to-teleport of 120 s. We had to make the decision to put halting buses in parking mode while servicing bus stops so that they do not block roads while halting. Most of the time, this assumption is reasonable as halting bays are rarely modeled in OSM and therefore cannot be included in the BeST network and also opposite edge overtaking is difficult to model in SUMO. Table 2 shows an excerpt of the simulation setup. Both scenarios are run three times using different random seeds to achieve significant results, with all simulation durations in the range of seven hours on an Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20 GHz CPUs using SUMO version 1.22.0.

4.2 General Statistics

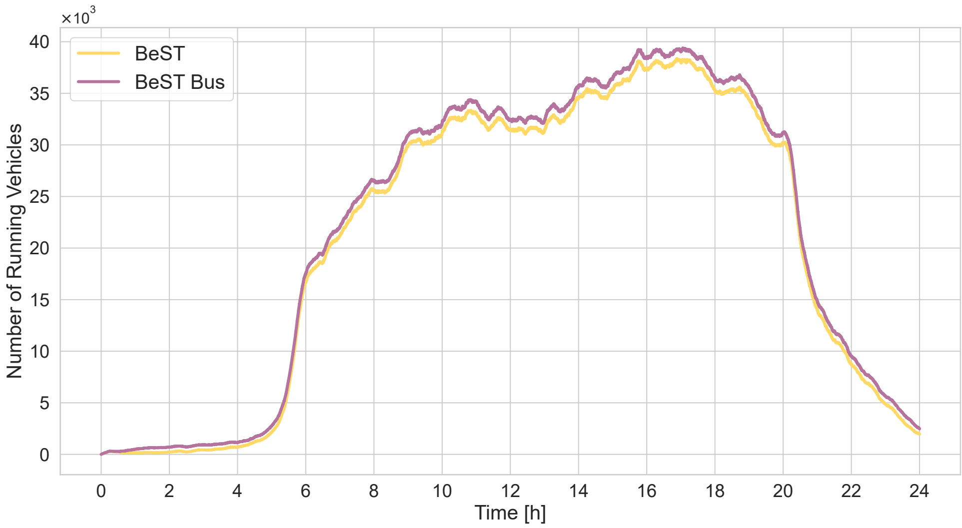


Figure 7. *Comparison of running vehicles between BeST and BeST Bus scenarios.*

Figure 7 illustrates the number of running vehicles over 24 h, both for the BeST and the BeST Bus scenario. It can be seen that bus traffic almost adds a constant offset to the overall traffic volume, but remains a small fraction of approximately 1 %.

In both scenarios, morning and evening peaks occur around 9 a.m. and 5 p.m., respectively, indicating a realistic traffic pattern. To gather initial insights into the impact of added bus traffic, various simulation-level metrics such as teleports, simulation time, and others are considered. The results are summarized in Table 3.

While the total number of vehicles increases only by less than 1 %, teleports increase by about 11.57 %, from around 0.93 teleports per 1000 vehicles to 1.04. In the BeST Bus scenario, 93 buses were teleported which is equivalent to 3.75 teleports per 1000 buses, a number three times higher than the total. Although these numbers show an increase in traffic disruption, the overall effect is within limits, and bus traffic often does not affect background traffic in a meaningful way.

Table 3. *Summary of Average Simulation Results of the BeST Scenarios.*

Metric	BeST scenario	BeST Bus scenario
Total Teleports	2090	2332
Bus Teleports	-	93
Emergency Brakings	1074	1104
Average Trip Distance	7.86 km	7.90 km
Average Trip Duration	13.24 min	13.62 min
Average Trip Speed	33.73 km/h	33.37 km/h
Average Speed Performance Index	0.807	0.803

The average trip distances had a minor increase, likely due to bus routes typically ranging over larger distances. Almost consequently, general traffic metrics, such as average trip durations and speeds, and the Speed Performance Index (SPI), decreased slightly but mostly insignificantly.

In an effort to further investigate the impact of buses on car traffic, we separately analyze vehicle types, summarized in Table 4.

Table 4. *Comparison of regular vehicles and buses in the BeST Bus scenario.*

Metric	Passenger Vehicles	Buses
Average Trip Distance	7.86 km	11.96 km
Average Trip Duration	13.31 min	26.23 min (excl. stop times)
Average Trip Speed	33.54 km/h	27.53 km/h

Compared to the original scenario, cars experience a 6 s increase in travel time on average. The average speed decreases by 0.21 km/h. Although the impact of bus traffic is noticeable, the evaluated metrics indicate only a small impact on an individual scale, suggesting that the additional bus traffic causes some disruptions, but does not significantly affect overall car traffic performance.

4.3 On-Time Performance

To verify timeliness and schedule adherence of buses, the on-time performance (OTP) is evaluated. A bus is considered on time if it departs from a certain bus stop not more than 60 s behind schedule. As reference, we use the extracted GTFS schedules.

Applying the definition to all bus lines yields an on-time performance of 74.17 %, indicating that 74.17 % of stops have delays of 60 s or less. The average delay for all buses is 65 s, with a large standard deviation of 139 s.

Figure 8 presents the distribution of the average delays of all bus lines within the 95th percentile in a histogram. Using the OTP definition (indicated by the red line), 59.30 % of bus lines fall below the 60 s threshold on average (including all stops). However, outliers exist with one line showing an average delay of 10 min per stop, which requires manual inspection.

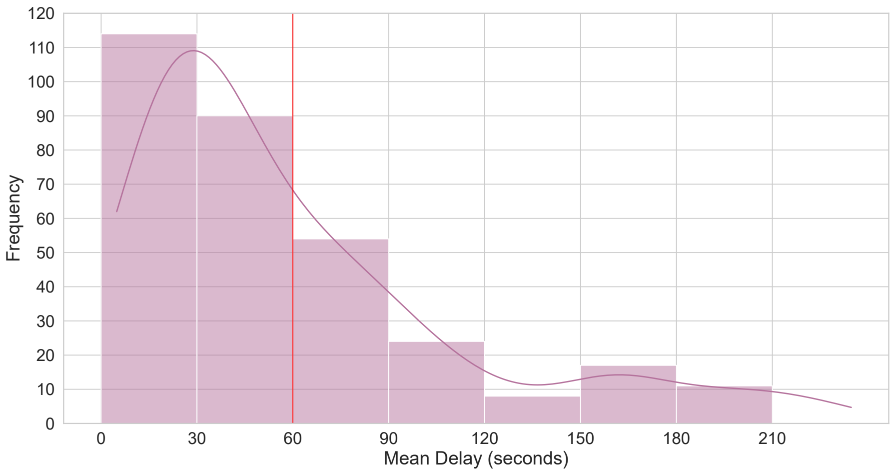


Figure 8. *Histogram of delays for the* 95*th percentile of entries.*

To evaluate a correlation between delays and general traffic conditions, we examine the average delays during the 24 h simulation period. Figure 9 shows the average delay for the 50 most-serviced bus lines (i.e., having the most trips). In addition, the average delay for all lines is indicated in black, serving as a reference to observe fluctuations in delay patterns. Compared to Figure 7, a slight correlation with the morning peak and a more prominent correlation with the evening peak at around 5 p.m. can be observed. During the period between 5 a.m. and 8 p.m., the variability per line route increases significantly, with notable outliers. Apart from the outliers, these results are to be expected. Larger delays and higher variability during peak hours are also observed with real buses, indicating a sensible integration with the prevalent traffic. While sparsely observed, this plot can also indicate times when bus-bunching may occur.

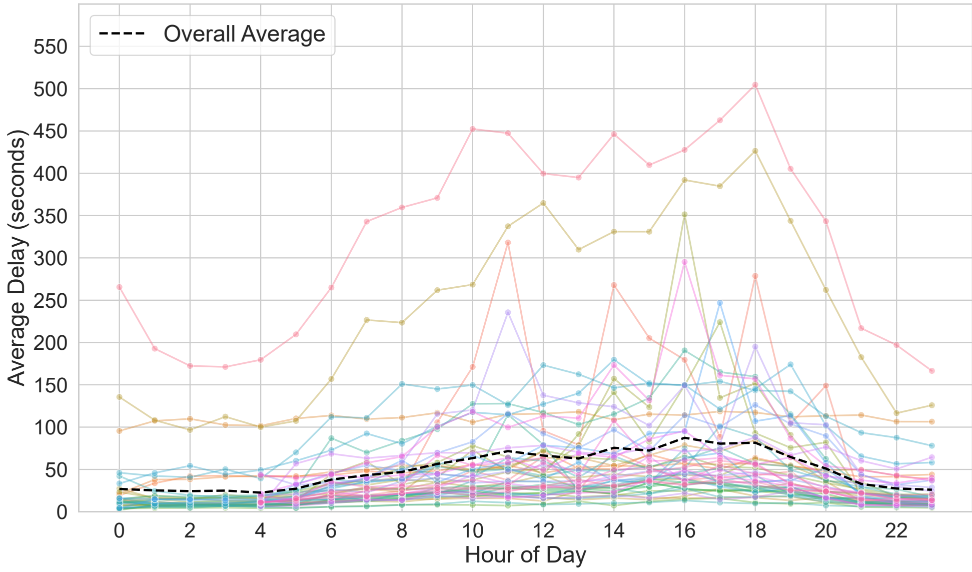


Figure 9. *Average delays throughout the day of the* 50 *most serviced bus lines*.

OTP data published by the BVG [12] claims that 88 % are on time. However, the BVG considers that buses with a delay of 3.5 minutes or less are on time. When applying this definition to the BeST Bus scenario, we would achieve an overall OTP of 92.24 %. Hence, we managed a successful integration of the underlying bus schedules.

4.4 Summary

The evaluation of the scenarios demonstrates that a stable BeST Bus scenario can be achieved. Key metrics, such as the total number of teleports and emergency brakings, show a measurable impact of bus traffic on overall simulation performance, although the effect remains minimal. The causes behind the relatively high number of bus teleports, particularly in relation to the total number of simulated buses, alongside the increase in yielding teleports, warrant further investigation. When isolating car traffic in the bus scenario, there is only minimal performance deterioration, with average speeds decreasing slightly and a modest six-second delay in average travel time. Using the strict OTP definition indicates that around 75 % of all stops are serviced on time.

5. Conclusion and Outlook

In this work, we presented an approach to integrate public transport into existing large-scale traffic scenarios. This approach is based solely on open data, such as OSM and open GTFS data. Most notable, a paradigm for map-matching and aligning GPS traces from GTFS shapes has been implemented. As a result, a second iteration of the Berlin SUMO Traffic (BeST) scenario was created, namely the Berlin SUMO Traffic plus Bus scenario (BeST-Bus). This scenario models 344 out of 386 bus lines and includes 6246 bus stops, resulting in 24,780 individual bus trips. A generalized schedule was extracted using the median period throughout the day for each line.

Unfortunately, it was not possible to automate the entire process. Especially, network extensions proved to be challenging. Clogging at junctions, missing edges, and other concerns demanded manual optimizations. The same goes for the introduction of bus lanes. Although we were able to automate the localization of missing infrastructure, merging of existing and new networks causes difficulties. As a proof-of-concept, we manually integrated bus lanes into the district of Charlottenburg, allowing for smaller-scale evaluations.

Our experiments show that adding buses disrupts existing vehicular traffic, but within acceptable thresholds. A re-calibration from the original demand, with added buses, could better the traffic throughout the network and relieve strain on bottlenecks.

Nonetheless, the resulting scenario is of great value for future research on transportation planning and novel ITS solutions, providing a baseline for experimentation. Specifically, our goal to use the scenario for research of fleet-provided traffic state estimation. In addition, further improvements are planned and ongoing. Since the private car traffic from the original scenario was generated using dynamic user assignment on an empty network, vehicles will potentially have chosen routes that are heavily affected by buses. Finally, inspection of unmappable lines revealed missing streets and errors in the BeST network, which were in parts manually adjusted.

Author contributions

Moritz Schweppenhäuser: Conceptualization, Methodology, Software, Writing - Original Draft, Writing - Review & Editing, Visualization
Timo Großmann: Methodology, Software, Writing - Original Draft, Writing - Review & Editing, Visualization
Karl Schrab: Software, Writing - Review & Editing, Supervision
Robert Protzmann: Writing - Review & Editing, Supervision
Ilja Radusch: Supervision

Competing interests

The authors declare that they have no competing interests.

References

[1] K. Schrab et al., “Modeling an ITS Management Solution for Mixed Highway Traffic With Eclipse MOSAIC,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 6, pp. 6575–6585, Sep. 23, 2023, ISSN: 1558-0016. DOI: 10.1109/TITS.2022.3204174.

[2] R. Protzmann, K. Schrab, M. Schweppenhäuser, and I. Radusch, “Implementation of a Perception Module for Smart Mobility Applications in Eclipse Mosaic,” in *SUMO Conference Proceedings*, vol. 3, 2022, pp. 199–214. DOI: 10.52825/scp.v3i.123.

[3] M. Schweppenhäuser, K. Schrab, R. Protzmann, and I. Radusch, “Evaluating spatiotemporal speed metrics for traffic state estimation on complex urban roads,” *SIMULATION*, p. 00 375 497 241 308 890, Jan. 23, 2025, ISSN: 0037-5497. DOI: 10.1177/00375497241308890.

[4] P. A. Lopez et al., “Microscopic Traffic Simulation Using SUMO,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, pp. 2575–2582. DOI: 10.1109/ITSC.2018.8569938.

[5] L. Codeca, R. Frank, S. Faye, and T. Engel, "Luxembourg SUMO Traffic (LuST) Scenario: Traffic Demand Evaluation," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 2, pp. 52–63, 2017, ISSN: 1941-1197. DOI: 10.1109/MITS.2017.2666585.

[6] L. Codecá and J. Härri, "Towards multimodal mobility simulation of C-ITS: The Monaco SUMO traffic scenario," in *2017 IEEE Vehicular Networking Conference (VNC)*, Nov. 2017, pp. 97–100. DOI: 10.1109/VNC.2017.8275627.

[7] S. C. Lobo, S. Neumeier, E. M. G. Fernandez, and C. Facchi, "InTAS – The Ingolstadt Traffic Scenario for SUMO," *CoRR*, Nov. 24, 2020. DOI: 10.48550/arXiv.2011.11995, pre-published.

[8] C. Gawron, "An Iterative Algorithm to Determine the Dynamic User Equilibrium in a Traffic Simulation Model," *International Journal of Modern Physics C*, vol. 09, no. 03, pp. 393–407, May 1998, ISSN: 0129-1831. DOI: 10.1142/S0129183198000303.

[9] K. Schrab, R. Protzmann, and I. Radusch, "A Large-Scale Traffic Scenario of Berlin for Evaluating Smart Mobility Applications," in *Conference on Sustainable Urban Mobility*, Springer, 2022, pp. 276–287. DOI: 10.1007/978-3-031-23721-8_24.

[10] OpenStreetMap contributors, *Planet dump retrieved from https://planet.osm.org*, https://www.openstreetmap.org, 2017.

[11] "GTFS Static Overview," Google for Developers, Accessed: Feb. 21, 2025. [Online]. Available: https://developers.google.com/transit/gtfs.

[12] "Parlamentsdokumentation (PARDOK) des Abgeordnetenhauses von Berlin," Accessed: Feb. 21, 2025. [Online]. Available: https://pardok.parlament-berlin.de/starweb/adis/citat/VT/19/SchrAnfr/S19-21316.pdf.