General Transit Feed Specification (GTFS) – And GTFS-Realtime Feed for Inner West Light Rail

Consumer Guide

People and Corporate Services

Systems

Contents

Change Record and Documentation Control

Version	Date	Name	Change Reference
0.5	26-Nov-2014	David Phillips	Initial version
0.6	16-Mar-2016	David Phillips	Updated for Open Data
0.7	21-July-2016	David Phillips	Enhancements to GTFS bundle
1.0	3-Aug-2016	David Phillips	Minor amendments following review
1.1	3-Oct-2025	Patrick Murphy	Updated for version 2.0 Feed

Reviewed By

Version	Date	Name	Role
0.7	29-July-2016	Stephanie Manning	Manager, Data Enablement, DCIS, CSD
0.7	1-Aug-2016	Ashish Ghimire	Quality Assurance Manager, Group IT
1.1	13-Oct-2025	Rose Firmin	Snr Product Manager, Customer Info Data, PIP

Author: David Phillips

Date: October 2025
Version: 1.1
Reference:
Division: Planning, Integration and Passenger
Review date:

Related Documents

Ref	Document Name	Version / Date	Location
01	GTFS Guides on Google Developers	Accessed 26-Oct-2015	https://developers.google.com/transit/gtfs/?hl=en
02	Realtime Transit on Google Developers	Accessed 26-Oct-2015	https://developers.google.com/transit/gtfs-realtime/
03	GTFS & GTFS R - TfNSW Implementation Specification V1.03	Accessed 7-Oct-2025	https://opendata.transport.nsw.gov.au/sites/default/files/2023-08/TfNSW_GTFS_GTFS-R__Implementation_Specification.pdf

1 Introduction

1.1 Overview

The purpose of this document is to describe the structure and contents of the Inner West Light Rail Timetable data feed supplied by Transdev and Lynxx for consumption and use by Transport for NSW systems and business units and application developers.

The Inner West Light Rail Timetable data feed is in the form of a General Transit Feed Specification fileset, known as 'GTFS'.

The intended audience of this document is application developers.

While GTFS is well documented, it provides several optional fields and some flexibility in how to populate some fields. This document provides some general information regarding the specific contents and structure of the GTFS Timetable Feed for Inner West Light Rail

1.2 Purpose of Sydney Light Rail GTFS Fileset

The purpose of the Inner West Light Rail GTFS Timetable Feed is to publish in advance the schedules and route information of Light Rail services on the L1 Dulwich Hill Line.

Consumers of the Inner West Light Rail GTFS Timetable Feed are expected to include:

- TfNSW Transport Info (http://www.transportnsw.info), Open Travel App.
- Realtime transport app providers (http://www.transportnsw.info/en/travelling-withus/keep-updated/apps/real-time-transport.page
- Open Data users (https://opendata.transport.nsw.gov.au)
- 3rd party apps such as Google Maps, TripView, NextThere and AnyTrip

1.3 Definition of Terms Used

Term	Definition
GTFS	The **General Transit Feed Specification** (GTFS) defines a common format for public transportation schedules and associated geographic information.
GTFS-R	**GTFS-realtime** is a feed specification that allows public transportation agencies to provide realtime updates about their fleet to application developers. It is an extension to GTFS.
IWLR	Inner West Light Rail services provided by Transdev Sydney Pty. Ltd.
Operator	Business contracted by TfNSW to operate Light Rail Routes
TransDev	Operator of the Inner West Light Rail service
Lynxx	Developer of the GTFS-R feed for the Inner West Light Rail service
Trip ID	A globally unique identifier for operator trips derived as part of the SIRI Broker transformation process. Provides the unique 'Trip ID' used in the GTFS Feed, and provides the link to GTFS Real-time data
SCADA	Supervisory Control and Data Acquisition, system used to monitor and control siginaling, tractions and other processes. Integrated into operational control of the Inner West Light Rail.
Dead Running	Non-passenger running of vehicles, usually for the purposes of position to a start of trip, or travelling to or from a depot.
TRACCAR	GPS tracking software used for the Inner West Light Rail v2 feed

2 Background

2.1 Lifecycle of GTFS Light Rail Timetable Feed

A brief overview of the process to generate the GTFS Light Rail Timetable feed.

- Sydney Light Rail operates and manages their service timetable in their source of truth system (HASTUS).
- When a change to service timetables is required the relevant updates are managed within HASTUS and a GTFS export process is initiated to generate the required GTFS data feed for delivery to Transport (via the GTFS Server). This process will not require Transport's intervention.
- File sets are transferred to the Open Data Portal via internal processes.

3 General Technical Notes

3.1 GTF Specification Compliance

3.1.1 Variations

The GTFS S bundle conforms to the GTFS & GTFS R – TfNSW Implementation Specification V1.0.3

3.1.2 Extensions

The GTFS S bundle conforms to the GTFS & GTFS R – TfNSW Implementation Specification V1.0.3

3.2 Publication Cycle & Scope

3.2.1 Validity Period and Timetable Projection

Fileset will contain:

- timetables commencing the day of generation
- a minimum of 100 days of timetables

3.2.2 Generation Triggers

GTFS Filesets are generated to reflect service changes on the Light Rail Network on an ad hoc basis

3.3 Standards Applied

3.3.1 Identifiers

Many of the identifiers used within the feed include dashed (-) to join elements. Care may need to be taken to consider this when using the data or developing applications.

3.3.2 Value Quoting

All values in the text files are double-quoted. This includes numeric values and empty values.

For example (from agency.txt):

```
"agency_id","agency_name","agency_url","agency_timezone","agency_lang","agency_phone"
,"agency_fare_url","agency_email"
"LR","Sydney Light
Rail","http://transportnsw.info","Australia/Sydney","EN","131500","",""
```

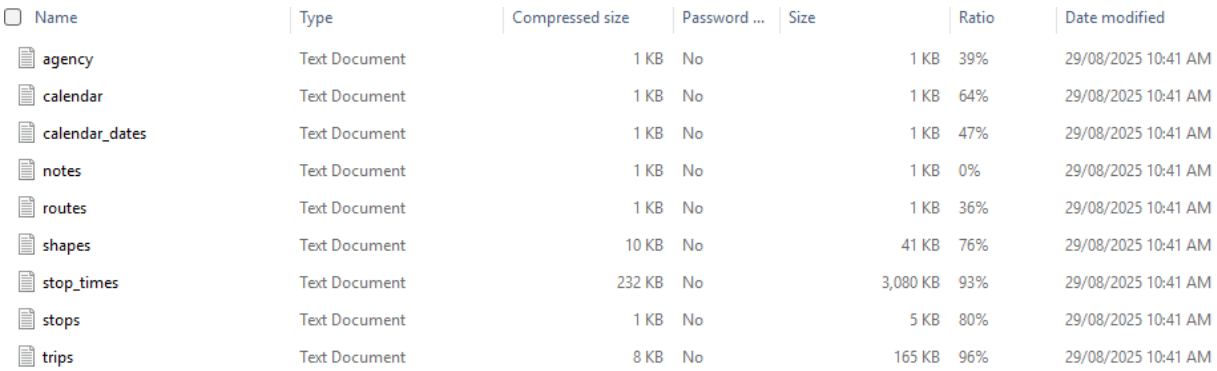
3.4 File Structure

3.4.1 Single Fileset Contents:

Each fileset will be a ‘ZIP’ format compressed archive – a .zip file - containing 9 text files.

Each file within the .zip file is a comma delimited / comma separated format file with the “.txt” extension.

For example:



Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
agency	Text Document	1 KB	No	1 KB	39%	29/08/2025 10:41 AM
calendar	Text Document	1 KB	No	1 KB	64%	29/08/2025 10:41 AM
calendar_dates	Text Document	1 KB	No	1 KB	47%	29/08/2025 10:41 AM
notes	Text Document	1 KB	No	1 KB	0%	29/08/2025 10:41 AM
routes	Text Document	1 KB	No	1 KB	36%	29/08/2025 10:41 AM
shapes	Text Document	10 KB	No	41 KB	76%	29/08/2025 10:41 AM
stop_times	Text Document	232 KB	No	3,080 KB	93%	29/08/2025 10:41 AM
stops	Text Document	1 KB	No	5 KB	80%	29/08/2025 10:41 AM
trips	Text Document	8 KB	No	165 KB	96%	29/08/2025 10:41 AM

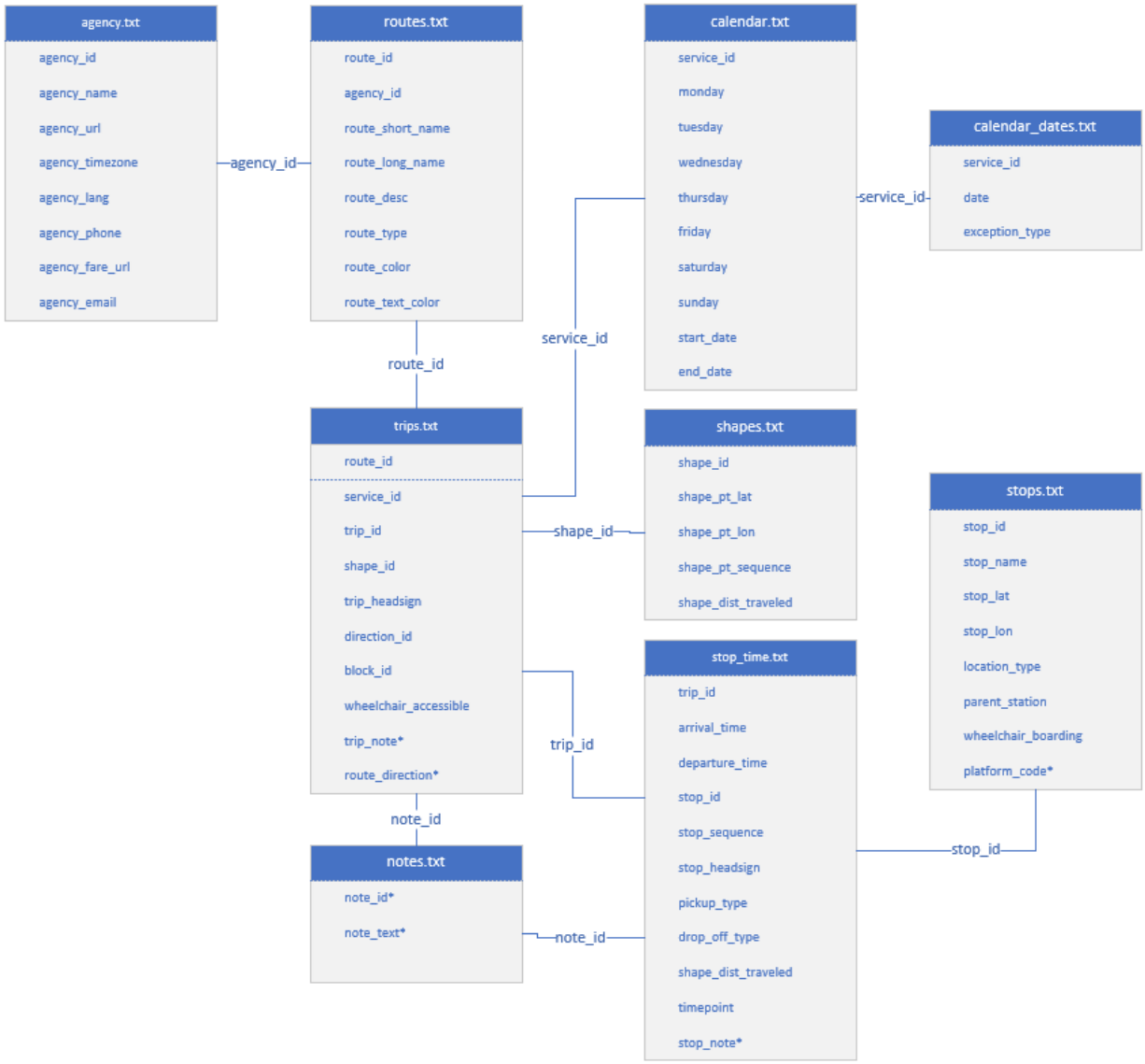
3.5 File Size

Compressed (.zip) files range in size from 300KB to 2MB. Files are unlikely to be larger than 10MB

4 GTFS Timetable Feed for NSW Light Rail

4.1 Overview

The following diagram provides a physical data model view of the relationship between the TfNSW GTFS Timetable Fileset files and fields.



Understanding this diagram:

- This is a representation of the relationships between the components of the GTFS Timetable fileset as implemented by TfNSW.
- "*" denotes a TfNSW implemented extension on the GTFS standard.
- GTFS files and columns not implemented by TfNSW are not shown.

4.2 agency.txt

Defines one or more transit agencies (Operators) that provide the data in this feed.

Field	Included	TfNSW GTFS Definition
agency_id	**Yes**	The Agency ID, allocated by TfNSW. For Example: "**LR**"
agency_name	**Yes**	The Operator's 'Customer Facing Name'. Note that this is not necessarily the same as the legal entity name. For example: "**Sydney Light Rail**"
agency_url	**Yes**	For all: "**http://transportnsw.info**"
agency_timezone	**Yes**	For all: "**Australia/Sydney**"
agency_lang	**Yes**	For all: "**EN**"
agency_phone	**Yes**	For all: "**131500**"
agency_fare_url	**No**	N/A
agency_email	**No**	N/A

4.3 calendar_dates.txt

Defines exceptions for the service IDs defined in the calendar.txt file.

Field	Included	TfNSW GTFS Definition
service_id	**Yes**	The internally generated ID that identifies a set of dates when a service exception is available for one or more routes. This identifier will be unique only within a GTFS Fileset, for example for a single Contract. This number has no meaning outside the Fileset.
date	**Yes**	The date field specifies a particular date when service availability is different than the norm. For Example: “20160315”
exception_type	**Yes**	Indicates whether service is available on the date specified in the date field. A value of 1 indicates that service has been added for the specified date. A value of 2 indicates that service has been removed for the specified date.

4.4 calendar.txt

Dates for service IDs using a weekly schedule. Specify when service starts and ends, as well as days of the week where service is available.

Field	Included	TfNSW GTFS Definition
service_id	**Yes**	The service_id contains an ID that uniquely identifies a set of dates when service is available for one or more routes. This identifier will be unique only within a GTFS Fileset, for example for a single Contract. This number has no meaning outside the Fileset.
monday	**Yes**	Each day field contains a binary value that indicates whether the service is valid for all of that day. A value of 1 indicates that service is available for all of that day in the date range. (The date range is specified using the start_date and end_date fields.) A value of 0 indicates that service is not available on that day in the date range. Note: Exceptions for particular dates, such as public holidays and service differences between school term and non-school term, are represented for the service_id in the calendar_dates.txt file.
tuesday	**Yes**	
wednesday	**Yes**	
thursday	**Yes**	
friday	**Yes**	
saturday	**Yes**	
sunday	**Yes**	
start_date	**Yes**	The start_date field contains the start date for the service.
end_date	**Yes**	The end_date field contains the end date for the service. This date is included in the service interval.

4.5 routes.txt

Transit routes. A route is a group of trips that are displayed to customers as a single service.

Field	Included	TfNSW GTFS Definition
route_id	**Yes**	The route_id field contains an ID that uniquely identifies a route. The route_id is dataset unique. For example: "**IWLR-191**"
agency_id	**Yes**	The TfNSW agency ID, allocated by TfNSW. For Example: "**LR**"
route_short_name	**Yes**	The short code identifying the Route to the public. For example: "**L1**"
route_long_name	**Yes**	The long name identifying the Route to the public. For example: "**Dulwich Hill Line**"
route_desc	**Yes**	***Note: TfNSW Variation.*** Indicates the Transport Network that the Route belongs to, as defined by **TfNSW** For example: "**Sydney Light Rail Network**".
route_type	**Yes**	Describes the type of transportation used on a route "**0**": Tram, Streetcar, Light Rail
route_url	**No**	N/A
route_color	**Yes**	**"EE343F"**
route_text_color	**Yes**	**"FFFFFF"**

4.6 shapes.txt

Rules for drawing lines on a map to represent a transit organization's routes.

Field	Included	TfNSW GTFS Definition
shape_id	**Yes**	The shape_id field contains an ID that uniquely identifies a shape.
shape_pt_lat	**Yes**	These fields associate a shape point's longitude and latitude with a shape ID. The field values must be valid WGS 84 values from -180 to 180. Each row in shapes.txt represents a shape point in the trip's shape definition. For example: "**-32.8407**","**151.3551139**"
shape_pt_lon	**Yes**	
shape_pt_sequence	**Yes**	The shape_pt_sequence field associates the latitude and longitude of a shape point with its sequence order along the shape. The values for shape_pt_sequence will be non-negative integers, and they will increase along the trip. These numbers have no meaning outside the trip point sequence.
shape_dist_traveled	**Yes**	The shape_dist_traveled field positions a shape point as a distance traveled along a shape from the first shape point. The shape_dist_traveled field represents a real distance traveled along the route in **meters**. The values used for shape_dist_traveled will increase along with shape_pt_sequence: they cannot be used to show reverse travel along a route.

4.7 stop_times.txt

Times that a vehicle arrives at and departs from individual stops for each trip.

Field	Included	TfNSW GTFS Definition
trip_id	**Yes**	The trip_id field contains an ID that identifies a trip. This value is referenced from the trips.txt file. This ID is internally generated by the Light Rail service provider.
arrival_time	**Yes**	The arrival time at a specific stop for a specific trip on a route. Times for trips starting before 04:00 am will be expressed in '36 hour format'. For example: "**25:07**" (01:07 am)
departure_time	**Yes**	The departure time from a specific stop for a specific trip on a route. Times for trips starting before 04:00 am will be expressed in '36 hour format'. For example: "**25:09**" (01:09 am)
stop_id	**Yes**	The TSN ID for the Stop. This uniquely identifies a stop. This ID is the unique Transit Stop Number that identifies a stop in the **Transit Stop Management System ('TSM')**, the original Transit Stop reference resource maintained by TfNSW. For example: "**2000257**"
stop_sequence	**Yes**	The sequence of the Stop within the Trip. The stop_sequence will be a non-negative integer, and will increase along the trip. For example: "**3**" indicates that the stop is the 3rd in the Trip.
stop_headsign	**Yes**	For all: "" (i.e. null)
pickup_type	**Yes**	Indicates whether the Stop is for Pickup. For example: "**0**" indicates that regularly scheduled pickup occurs at the Stop for the Trip.
drop_off_type	**Yes**	Indicates whether the Stop is for Pickup. For example: "**1**" indicates that no drop off available for the Stop for the Trip.
shape_dist_traveled	**Yes**	The shape_dist_traveled field positions a shape point as a distance traveled along a shape from the first shape point. The shape_dist_traveled field represents a real distance traveled along the route in **meters**. The values used for shape_dist_traveled will increase along with stop_sequence: they cannot be used to show reverse travel along a trip.
timepoint	**Yes**	Indicates whether the Stop is a timing point for the Trip. TfNSW consider all Light Rail stops to be timing points For example: "**1**" indicates that the Stop is a Timing Point for the Trip.

Field	Included	TfNSW GTFS Definition
stop_note	**Yes**	***Note: TfNSW Extension.*** To reflect Notes provided by Operators for each Stop, to be available to Customers. May be empty. The value is referenced from the notes.txt file. For example: "**2143**" refers to notes.txt note_txt "**Stops only on request**".

Additional notes:

- Arrival and departing times should be used verbatim from the GTFS Timetable Feed. **Refer Error! Reference source not found.**.

4.8 stops.txt

Individual locations where vehicles pick up or drop off passengers.

Field	Included	TfNSW GTFS Definition
stop_id	**Yes**	The Transit Stop Number (TSN) ID for the Stop. This uniquely identifies a stop. This ID is the unique TSN that identifies a stop in the **Transit Stop Management System (TSM)**, the original Transit Stop reference resource maintained by TfNSW. For example: "**2000257**"
stop_code	**No**	N/A
stop_name	**Yes**	The correct name of the Stop. This Stop name is the 'official' name for the Stop as recorded in the TSM System. For example: "**Central Station Light Rail**"
stop_desc	**No**	N/A
stop_lat	**Yes**	The WGS 84 longitude ('y' coordinate') for the stop. For example: "**-33.882378**"
stop_lon	**Yes**	The WGS 84 longitude ('x' coordinate') for the stop. For example: "**151.206724**"
zone_id	**No**	N/A
stop_url	**No**	N/A
location_type	**Yes**	For all: "**0**" Indicates that the stop is a 'Stop' as opposed to a 'Station'.
parent_station	**Yes**	For all: "" (i.e. null).
stop_timezone	**No**	N/A
wheelchair_boarding	**Yes**	For all: "**1**" Indicates that the stop is accessible
platform_code	**Yes**	Light Rail stop do not have platform numbers visible to customers. The GTFS bundle contains indicative values to show the direction of travel. "**1**": Towards Central "**2**": Towards Dulwich Hill

4.9 trips.txt

Trips for each route. A trip is a sequence of two or more stops that occurs at specific time.

Field	Included	TfNSW GTFS Definition
route_id	**Yes**	The route_id field contains an ID that uniquely identifies a route. The route_id is dataset unique. The value is referenced from route.txt. For example: "**IWLR-191**"
service_id	**Yes**	The service_id contains an ID that uniquely identifies a set of dates when service is available for one or more routes. This identifier will be unique only within a GTFS Fileset, for example for a single Contract. This number has no meaning outside the Fileset.
trip_id	**Yes**	The trip_id field contains an ID that identifies a trip. This value is referenced from the trips.txt file. This ID is internally generated by the Light Rail service provider
trip_headsign	**Yes**	The trip_headsign field contains the text that appears on a sign that identifies the trip's destination to passengers. For example: "**Central Station**"
trip_short_name	**No**	N/A
direction_id	**Yes**	Indicates the direction (outgoing, inbound) of the Trip. Possible values are: • "0": Outbound • "1": Inbound
block_id	**Yes**	Identifies the block to which the trip belongs. A block consists of two or more sequential trips made using the same vehicle, where a passenger can transfer from one trip to the next just by staying in the vehicle. Eg: "**P-1-729828**"
shape_id	**Yes**	Contains an ID that defines a shape for the trip. This value is referenced from the **shapes.txt** file.
wheelchair_accessible	**Yes**	Indicates wheelchair accessibility for a scheduled trip. All Light Rail trips are considered wheelchair accessible Possible values are: • "**1**": Accessible service.
bikes_allowed	**No**	• "**1**": Indicates that the vehicle being used on this particular trip can accommodate at least one bicycle
trip_note	**Yes**	***Note: TfNSW Extension.*** To reflect Notes provided by Operators for each Trip, to be available to Customers. May be empty.

Field	Included	TfNSW GTFS Definition
route_direction	**Yes**	***Note: TfNSW Extension.*** To describe the Route Direction for each Trip, to be visible to Customers in Timetables. Enables grouping of Trips together in a meaningful way. For example: "**Dulwich Hill to Central**"

4.10 notes.txt

This file is an extension on the GTFS Fileset standard.

Contains a list of notes referenced from trips.txt and stop_times.txt. At the time of publication this file is unpopulated.

Field	Included	TfNSW GTFS Definition
note_id	**Yes**	Unique ID for Notes referenced in trips.txt (trip_note) and stop_times.txt (stop_note).
note_txt	**Yes**	Text for note used in stop_times.txt and trips.txt. A single Note may be re-used multiple times.

5 GTFS Realtime

5.1 Overview

"GTFS-realtime is a feed specification that allows public transportation agencies to provide realtime updates about their fleet to application developers. It is an extension to GTFS (General Transit Feed Specification), an open data format for public transportation schedules and associated geographic information. GTFS-realtime was designed around ease of implementation, good GTFS interoperability and a focus on passenger information."

- 'What is GTFS-realtime', Google Realtime Transit Overview (https://developers.google.com/transit/gtfs-realtime/)

The TfNSW Light Rail GTFS Realtime feed developed by Lynxx is generated by TRACCAR positions, information from the Supervisory Control and Data Acquisition (SCADA) system used to monitor vehicle activity and GTFS Schedule data created by Transdev in their HASTUS system. Trip Update and Vehicle Position both use the proto buff format.

5.2 2. GTFS-R Trip updates

Trip updates use an allocation system combined with vehicle tracking data to populate the Trip Update feed. Blank allocations are generated every day, using vehicle position data in relation to static GTFS schedule information. Allocations are assigned to each vehicle on a trip before they start using vehicle speed, previous allocation, geofencing locations, allocations of other vehicles, manual vehicle deallocation derived from SCADA inputs and predicted arrival times of final stops to ensure accurate allocations in the schedule. Unallocated trips are set to cancelled through the predictive allocation system along side manual intervention from TransDev operators through their operational command centre (OCC.)

5.2.1 Actual Trip Updates

Actual trip updates are produced at events such as a vehicle entering or leaving a stops geofence.

5.2.2 Predictions

Predictions are produced periodically between stops, as well as at the time of events like departures and arrivals for current and upcoming trips. Scheduled data and historic predictions are used to propagate delays through multiple trips that the vehicle runs according to its block allocation. Predictions are also used to improve future allocations.

5.2.3 Allocation

Allocation works on a 'best fit' basis via vehicle commencing from the start of trip location (usually Central or Dulwich Hill) and assigning it to the closest appropriate blank allocation.

5.2.4 Elemental Index

Message Structure	Message Example

```
{
“header”: {
  “gtfs_realtime_version”:
  “incrementality”:
  “timestamp”:
}
“entity”: {
  “id”:"
  “trip_update”: {
    “trip”: {
      “trip_id”:,
      “start_time”:,
      “start_date”:,
      “schedule_relationship”:
      “route_id”:
      “direction_id”:
    }
    “stop_time_update”: {
      “stop_sequence”:
      “arrival”: {
        “delay”:
        “time”:
        “uncertainty”:

      }
      “departure”: {
        “delay”:
        “time”:
        “uncertainty”:
      }
      “stop_id”:"
      “schedule_relationship”:
    }
    “vehicle”: {
    “id”:
    }
    “timestamp”:
    “delay”:
  }
}
```

```
{
header {
  gtfs_realtime_version: "2.0"
  incrementality: FULL_DATASET
  timestamp: 1758598731
}
entity {
  id: "43588-1156"
  trip_update {
    trip {
      trip_id: "43588-1156"
      start_time: "14:38:00"
      start_date: "20250923"
      schedule_relationship: SCHEDULED
      route_id: "IWLR-191"
      direction_id: 0
    }
  stop_time_update {
      stop_sequence: 1
      arrival {
        delay: -5
        time: 1759381405
        uncertainty: 5
       }
      departure {
        delay: -5
        time: 1759381405
        uncertainty: 5
      }
      stop_id: "2000257"
      schedule_relationship: SCHEDULED
    }
    stop_time_update {
      stop_sequence: 2
      arrival {
        delay: -3
        time: 1759381498
        uncertainty: 3
       }
      departure {
        delay: -3
        time: 1759381498
        uncertainty: 3
     }
      stop_id: "2000256"
      schedule_relationship: SCHEDULED
      }
_________________Continues to stop 23_____________
    stop_time_update {
      stop_sequence: 23
      arrival {
        delay: -4
        time: 1759383626
        uncertainty: 4
      }
      departure {
        delay: -4
        time: 1759383626
        uncertainty: 4
      }
      stop_id: "220363"
      schedule_relationship: SCHEDULED
    }
```

	vehicle {
	id: "669989"
	}
	timestamp: 1759379877
	delay: -5
	}
	}

5.3 3. GTFS-R Vehicle Position

5.3.1 3.1 Coverage

Vehicle coverage is based on tracking devices installed within operational light rail vehicles utilising pinging the vehicle position every 3 seconds to an API endpoint. In areas of low signal, the feed goes into a ‘low signal mode’ covered in exception cases to predict most likely vehicle positions.

5.3.2 VehicleID

Vehicle ID will be the unique identifier of the beacon on the vehicle to account for the interchangeability of these devices not being locked to a specific vehicle. This is due to the mixed fleet structure for the L1 and potential use of vehicles from the L2 & L3.

5.3.3 Elemental Index

Message Structure	Message Example

{
header {
“gtfs_realtime_version”:
“incrementality”:
“timestamp”:
}
“entity”: {
“id”:
“vehicle”: {
“trip": {
“trip_id”:
“start_time”:
“start_date”:
“schedule_relationship”:
“route_id”:
“direction_id”:
}
“position”: {
“latitude”:
“longitude”:
“bearing”:
“speed”:
}
“current_stop_sequence”:
“current_status”:
“timestamp”:
“congestion_level”:
“stop_id”:
“vehicle”: {
“id”:
“[transit_realtime.tfnsw_vehicle_descriptor]”: {
“air_conditioned”:
“wheelchair_accessible”:
“vehicle_model”:
}
}
“[transit_realtime.consist]:” {
“position_in_consist”:
	{
	header {
	gtfs_realtime_version: "2.0"
	incrementality: FULL_DATASET
	timestamp: 1758608224
	}
	entity {
	id: "100943422"
	vehicle {
	trip {
	trip_id: "43588-1193"
	start_time: "16:22:00"
	start_date: "20250923"
	schedule_relationship: SCHEDULED
	route_id: "IWLR-191"
	direction_id: 0
	}
	position {
	latitude: -33.882267
	longitude: 151.20651
	bearing: 13.767521
	speed: 1.0145731
	}
	current_stop_sequence: 2
	current_status: IN_TRANSIT_TO
	timestamp: 1758608214
	congestion_level: UNKNOWN_CONGESTION_LEVEL
	stop_id: "2000256"
	vehicle {
	id: "118466"
	[transit_realtime.tfnsw_vehicle_descriptor] {
	air_conditioned: true
	wheelchair_accessible: 1
	vehicle_model: "Alstom Citadis X05 Light Rail Vehicle"
	}
	}
	[transit_realtime.consist] {
	position_in_consist: 1

“quiet_carriage”:
“toilet”:
“luggage_rack”:
}
}
}
	quiet_carriage: false
	toilet: NONE
	luggage_rack: false
	}
	[transit_realtime.consist] {
	position_in_consist: 2
	quiet_carriage: false
	toilet: NONE
	luggage_rack: false
	}
	}
	}
	entity {
	id: "100942975"
	vehicle {
	trip {
	trip_id: "43588-1187"
	start_time: "16:04:00"
	start_date: "20250923"
	schedule_relationship: SCHEDULED
	route_id: "IWLR-191"
	direction_id: 0
	}
	position {
	latitude: -33.8687
	longitude: 151.19623
	bearing: 320.14938
	speed: 9.881336
	}
	current_stop_sequence: 7
	current_status: IN_TRANSIT_TO
	timestamp: 1758608136
	congestion_level: UNKNOWN_CONGESTION_LEVEL
	stop_id: "200942"
	vehicle {
	id: "198295"
	[transit_realtime.tfnsw_vehicle_descriptor] {
	air_conditioned: true
	wheelchair_accessible: 1
	vehicle_model: "CAF Urbos 100 Light Rail Vehicle"
	}
	}
	[transit_realtime.consist] {
	position_in_consist: 1
	quiet_carriage: false
	toilet: NONE
	luggage_rack: false
	}
	[transit_realtime.consist] {
	position_in_consist: 2
	quiet_carriage: false
	toilet: NONE
	luggage_rack: false
	}
	}
	}

6 Exception Cases

6.1 Dead Running Vehicles

Dead-running vehicles are identified in 2 different ways.

6.1.1 Using SCADA Data

SCADA data contains dead-running information, which informs when a dead-running vehicle passes a particular stop. This data, along with vehicles near this stop, is used to identify the dead-running vehicle. This vehicle is then set on a ‘dead-running’ trip, which deallocates it from current trip and prevents it from allocating to future trips until it reaches another terminus.

6.1.2 Vehicle Movement

In addition, dead-running vehicles can be identified when they pass stops without stopping as an example of not passenger loading behaviour. Speed and positional data is analysed at key points along trips to dynamically deallocate trips so they can be allocated to other vehicles if applicable.

6.2 Auto-Cancellations

If a trip has not been allocated to a vehicle, but later trips have been allocated, it will be auto cancelled. This involves looking back when making allocation predictions, as order of predictions means there is no vehicle that could pick up this trip (unless one comes from a depot). This is reversed in the case where a depot vehicle attempts to allocate itself to these trips. These cancellations are not pushed to the feed until after the trip start timestamp.

6.3 Automatic Disruption Detection

If a vehicle turns around and travels in the opposite direction for a distance, the system will deduce that a disruption on the tracks has caused the reversal. In these situations, the feed skips the next stops of the trip for the remainder of the schedule it was on and creates another ADDED trip that goes in the opposite direction. The vehicle is allocated to this new ADDED trip.

6.4 Low Signal Handling

In the event of a low signal or a dropout drops out due to connectivity issues in sections of the track, the vehicle goes into ‘low signal’ mode. The system periodically checks whether any vehicle has not published its position for a period of time. When it detects a vehicle that has lost connection, it creates a propagated signal considering the last known position of the vehicle and the scheduled times. This propagated signal is also capable of stopping at stops in the same way the actual vehicle would, to simulate expected behaviour. Once signal has been restored, vehicle position adopts normal running behaviour.