Acyclica Go

User Guide

v3.5

Last modified November 22, 2017

Contents

Section 1

Overview

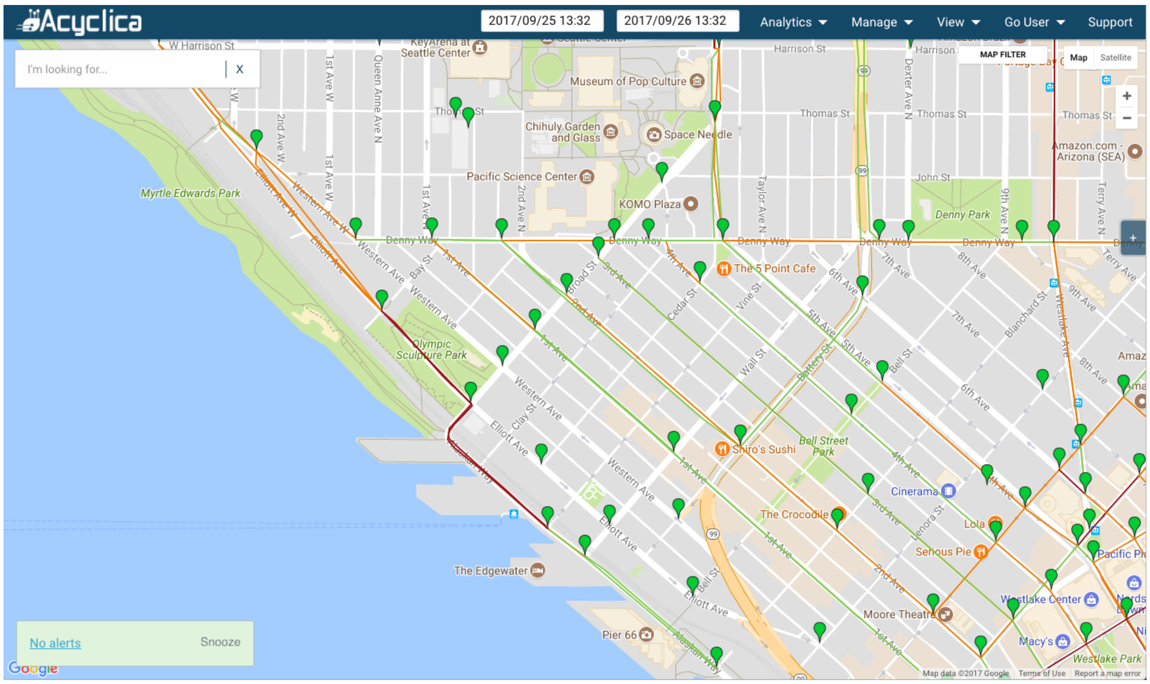

Figure 1.1 - Go Interface

1.1 Interface

Go's interface (Figure 1.1) is made up of the following components:

- *Top Nav*
  - Displayed along the top of the interface, this includes the Date Picker tools and dropdowns to view Analytics, Configuration Options, User Information, and Help
- *Map*
  - Displays Locations, Segments, Routes, and a visual representation of traffic data
- *Alerts*

- Located in the lower left corner, this panel will be green when there are no active alerts. When there are active alerts, it will be pink, and will contain the number of active alerts

- *Locations*
  - Indicated by markers on the map, a Location is the set of hardware that exist at a single physical location. A Location can hold one or more sensors from Acyclica and other manufacturers
  - There are 5 states:
    - Online (Green) - all equipment at a location is uploading data
    - Selected (Blue) - used when a location is selected
    - Offline (Gray) - no recent data uploaded from any hardware at a location
    - Some Hardware Offline (Split Green and Gray) - some hardware at a location has recently uploaded data, and some has not uploaded data recently
    - Alert (Red) - alert is in progress at a location

- *Information Panel*
  - Displayed when an item has been selected. This includes sensors, routes, and segments. It will show information about the item, analytics that can be run, and associated sensors or routes

1.2 Registration

To create an account for Acyclica Go:

1. Open https://go.acyclica.com in your web browser
2. Select **Create Your Account**, as seen at the bottom of Figure 1.2
3. Complete the registration form
4. Please note the password requirements:
   - Password must be eight (8) characters or longer
   - Password cannot be your username
   - Password cannot be in the list of 1000 most common passwords
5. Select **Sign Up**
6. Open the verification email sent to your address
7. Select the account verification link in the email

In order to see sensors, routes, and other information in Go, you must be part of a client group. You will already be part of a client group if you received an invitation to join and signed up with the email address that invitation was sent to. If you didn't receive an invitation, your client group administrator will need you to add you to the group. Please contact support@acyclica.com if you don't know who your client group admin is.

Figure 1.2 - Go Login

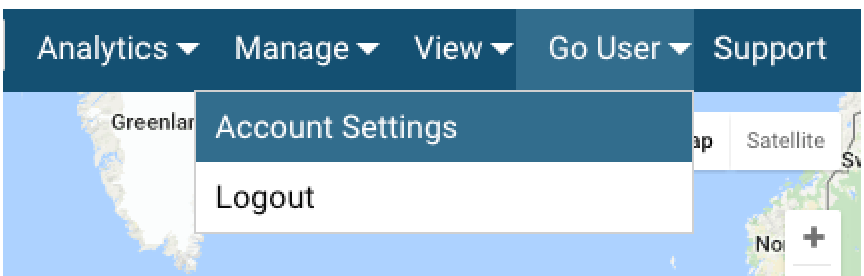

Figure 1.3 - Access Account Settings.

1.3 User Types

There are four types of users in Go:

- *Standard Users*
    - Can access all analytics and modules available to the client group, but cannot make any configuration changes
- *Configuration Admins*
    - All the same permissions as a Standard User, and also have the ability to configure hardware, create routes, create alerts, and so forth
- *User Admins*
    - Can add or remove users from their client groups, and set admin levels.
- *Client Group Owners*
    - Have complete access to all permissions available. Client Group Ownership privileges can only be added or removed by Acyclica Support

If you don't know who your client group admin is and need additional access, contact support@acyclica.com .

Chapter 2

User Management

2.1 Introduction

User Administrators can use the My Groups section of Settings to view lists of users, manage the users within their groups, and set the access privileges that those users have.

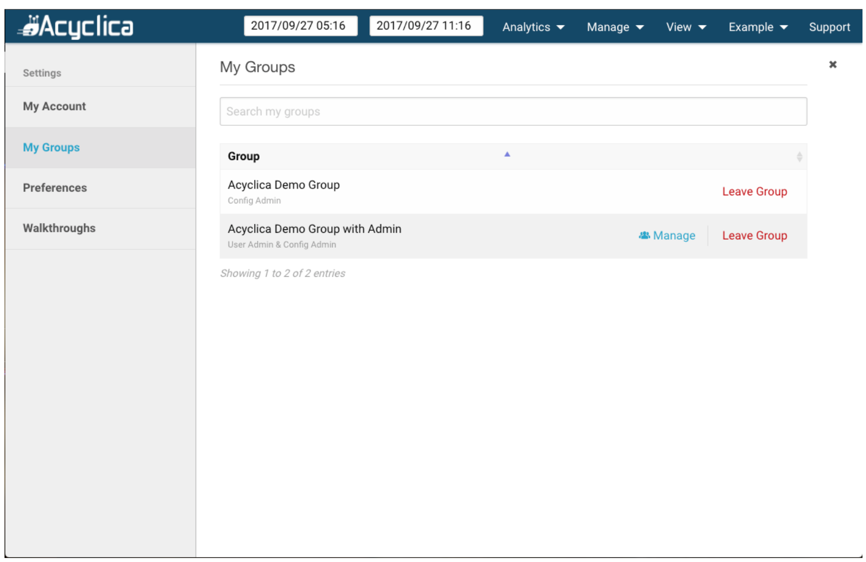

Figure 2.1 - List of client groups for the example user.

As seen in Figure 2.1, the Settings section for *My Groups* will list all of the client groups the user can access. There are four levels of user access to a client group, which are as follows:

- **Group Owners** can manage group memberships, permissions, setup sensors, and create routes, O/D Groups, and alerts; their permissions *cannot* be changed by user admins or other owners
- **User Admins** can manage group membership including adding or removing users and setting permissions
- **Config Admins** can setup sensors and create routes, O/D groups, and alerts
- **Standard Users** have access to run and view analytics in the software

Group Owners and User Admins have the ability to modify users' access to the client group, and can raise or lower a user's permissions within the client group.

2.2 Managing the Client Group

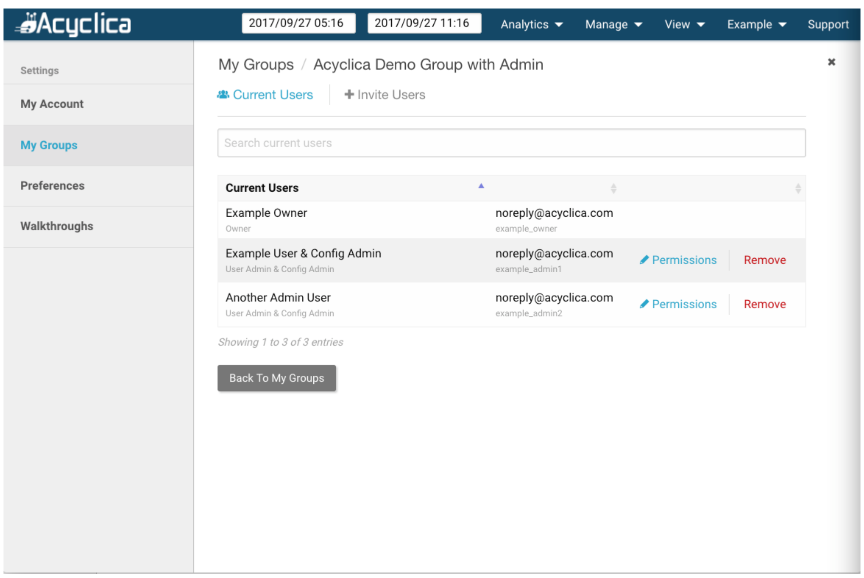

Figure 2.2 - List of users in an example client group.

Selecting **Manage** on the *My Groups* tab of the settings view will allow a user with User Admin or Group Owner to see a list of all non-owner users in the selected client group, as seen in Figure 2.2. From here, individual user permissions may be modified. Selecting **Invite Users** at the top will allow a user to invite additional users to the client group, as well as allowing the user to set those invited users' permission levels.

2.2.1 User Permissions

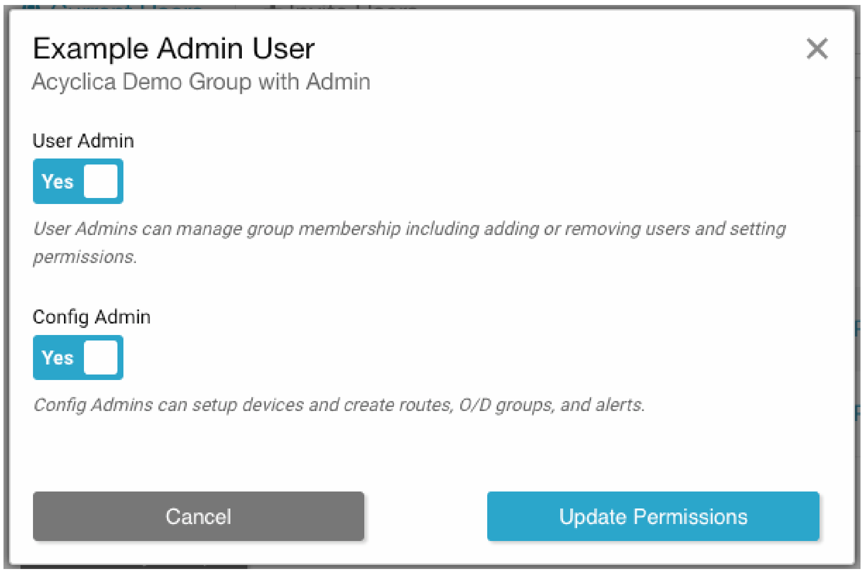

Figure 2.3 - Modifying user permissions.

An individual user's permissions are defined by what level of user access that user has to the client group, and is determined on a per-client group basis. For example, a user may be a User Admin in one group, but a standard user in another. This would allow the user to manage the other users in the first client group, but only allow that user to run and view analytics in their second client group.

2.2.2 Adding New Users

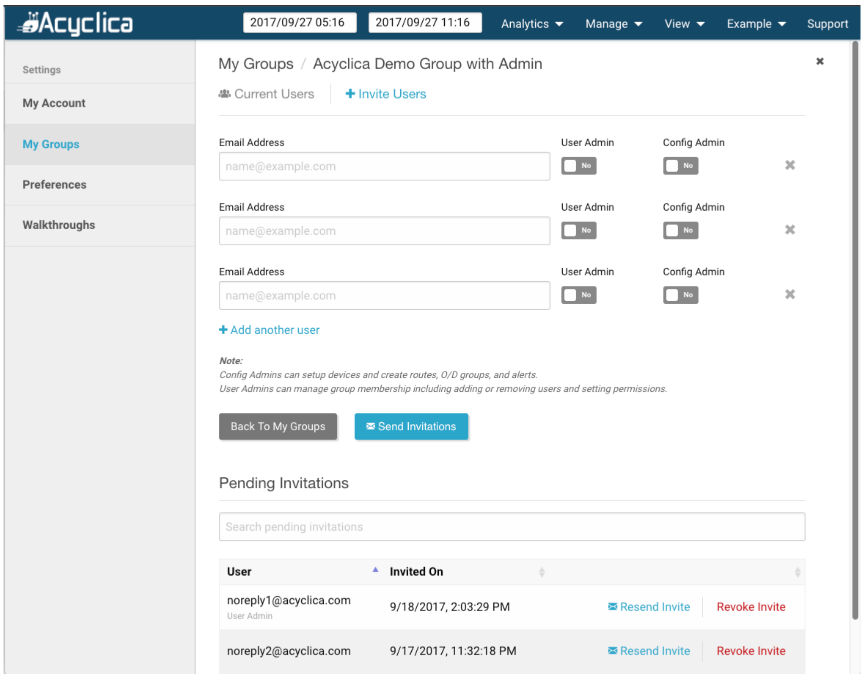

Figure 2.4 - Invite new users and manage user invitations.

The different levels of access that may be configured by a client group's Group Owner and User Admin level users are shown in Figure 2.3. Group Owners can only be set by Acyclica Support, please contact Support at support@acyclica.com . Figure 2.4 shows the **Invite Users** view, from which Group Owners and User Admins can manage new user invitations to their client groups.

Users are invited to a client group by entering their email address into the field, selecting their permission levels (any combination of User Admin, Config Admin, or neither), and selecting *Send Invitations*. Once the invitation is sent, the invited user will receive an email from noreply@acyclica.com prompting them to create an account, which will be automatically added to the group they are invited to join.

Section 3

Configuration

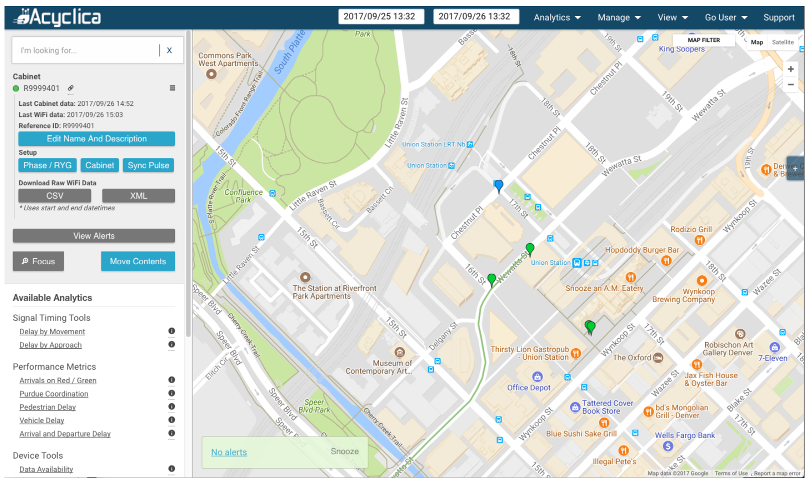

Figure 3.1 - Sensor configuration is accessed through the Information Panel.

3.1 Sensor

Sensor names, locations, and descriptions can be set in Go. This can be done before installation if the intended locations for all sensors are known, or can be done after installation in the field.

3.1.1 Name & Description

1. Select a location using the *Search Field* or by selecting the sensor on the map
    a) Note: if a name has not yet been given to the sensor, you can use the sensor's serial number in the search field.
2. Open the *hamburger menu* (three horizontal lines), next to the sensor to be edited
3. Select *Edit Name and Description*
4. Update as necessary
5. Select *Save*

Notes:

- The name is used across Go as a name for your sensor
- The short name is used for display in Locations where long sensor names may not fit
- The description field is useful as a general sensor notes area
- It is best to use a consistent naming convention for sensors. This makes it easier to use the search field to find sensors. If the name *Broadway* is used for some sensors and *Brdwy* is used for others, it can be more difficult to find those sensors with the search function. It is also a good idea to consistently name intersections, for example always using east-west before north-south.

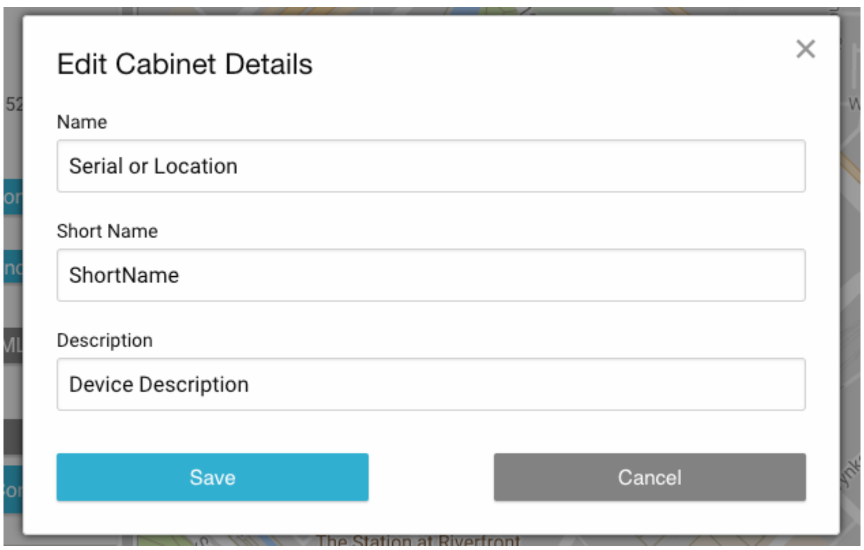

Figure 3.2 - Set the sensor's Name, Short Name, and Description.

3.1.2 Sensor Location

There are three methods for setting a sensor location:

- Entering Latitude and Longitude coordinates
- Selecting a location on the Map
- Via GPS positioning (Data Aggregators only)

To set a sensor location:

1. Select a sensor location using the search field or by locating it on the map
2. Select *Move Contents*
3. Set the location by one of the following methods:
   a) Enter Latitude and Longitude in the appropriate fields
   b) Select the desired location on the map
4. Select *Set Location*
5. If the sensor is part of one or more routes and it is being moved away from its currently location to a new one, it is recommended that you remove those routes to prevent data processing problems in the future. If it is a small tweak moving the sensor to a different spot at the same intersection this isn't necessary. A dialog will prompt you to do this:
   a) To delete routes, select *Delete Routes*
   b) To keep the routes, select *Keep Routes*

Notes:

- If a sensor is moved to a new location, the underlying connections that drive data processing will need to be updated by Acyclica. We will receive a notification that this needs to be done, and will complete the changes within one business day. If you need to include the sensor in routes sooner than that, please contact support and the changes will be performed as soon as possible
- Data Aggregators will be placed on the map automatically if there is a GPS signal and they have never been manually located in Go

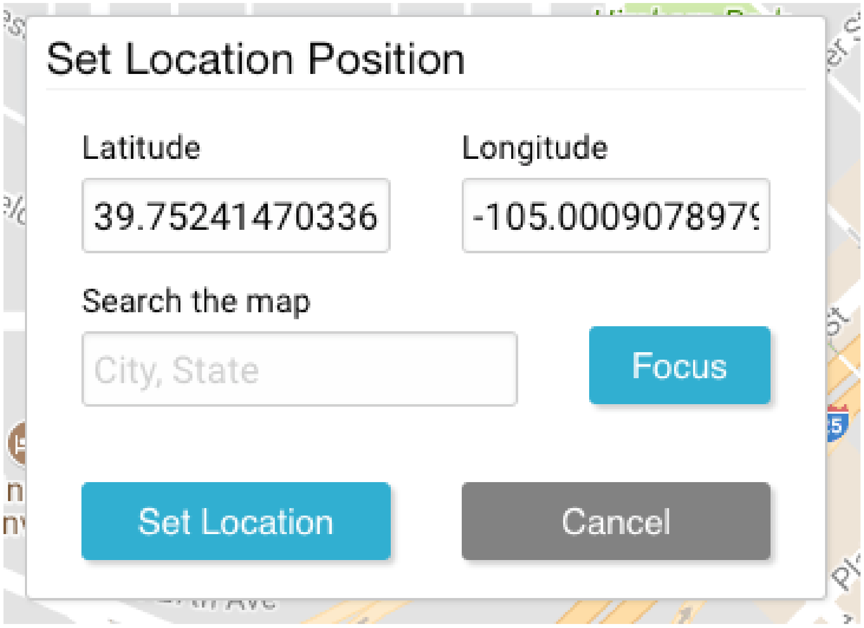

Figure 3.3 - Set the sensor's location.

3.2 Routes

3.2.1 Creating a Route

A route only represents traffic flow in one direction. To analyze traffic going both northbound and southbound, two different routes need to be created. Routes must include all sensors along the route. The Route Creation Tool now enforces this and will not allow you to select a sensor that is not adjacent to the last selected sensor. To create a route:

1. Hover over *Manage* in the Top Navigation Bar and select *Add Route*.
2. Select the sensors in the order that the route will be travelled. As sensors are selected, adjacent sensors will become available for your route (Figures 3.4, 3.5, and 3.6)
3. Once all of the sensors for the route have been added, select *Save* (Figure 3.7)
4. Set the route name
5. Select *Save*

Note: if you are attempting to create a route, and a sensor that you want to select is grayed out, the sensor connection definitions will need to be updated. To make this request, hover over the sensor you would like to use, and select the *Report* button.

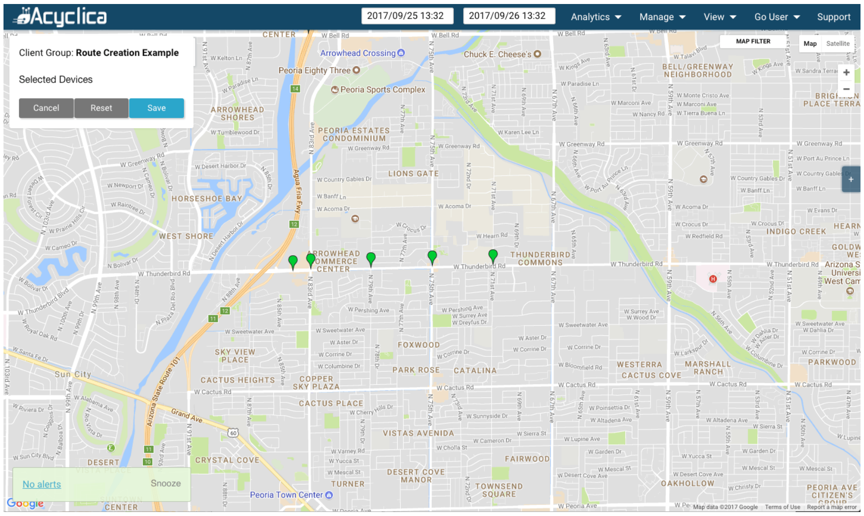

Figure 3.4 - All available sensors to start the route

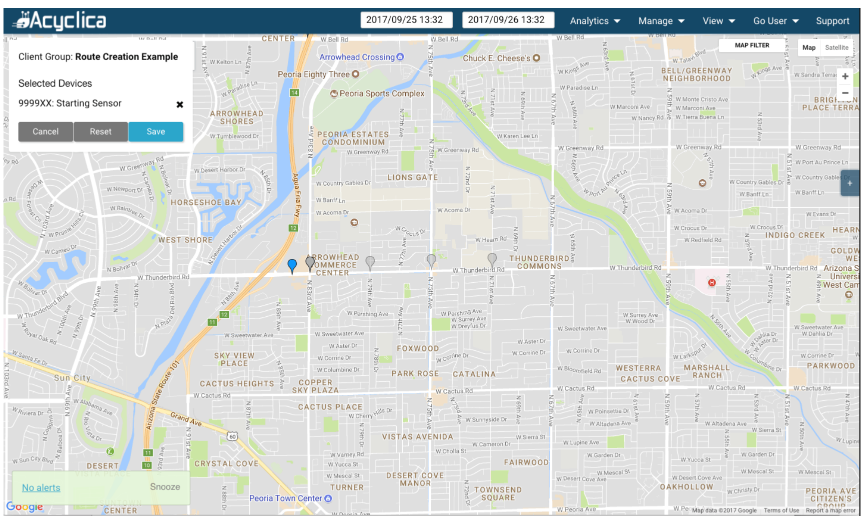

Figure 3.5 - First sensor selected, non-adjacent sensors are greyed out

3.2.2 Travel Time Alerts

Travel time alerts can be set on any created route. A travel time alert will send an email to your user when certain thresholds are exceeded. If the alert is set with a threshold of 120, triggers of 3 and time bins of 15, then this alert will trigger if there are is an average travel time of 120 seconds in three consecutive 15 minute spans. To create an alert, your user must have at least Config Admin permissions. These alerts can be created as follows:

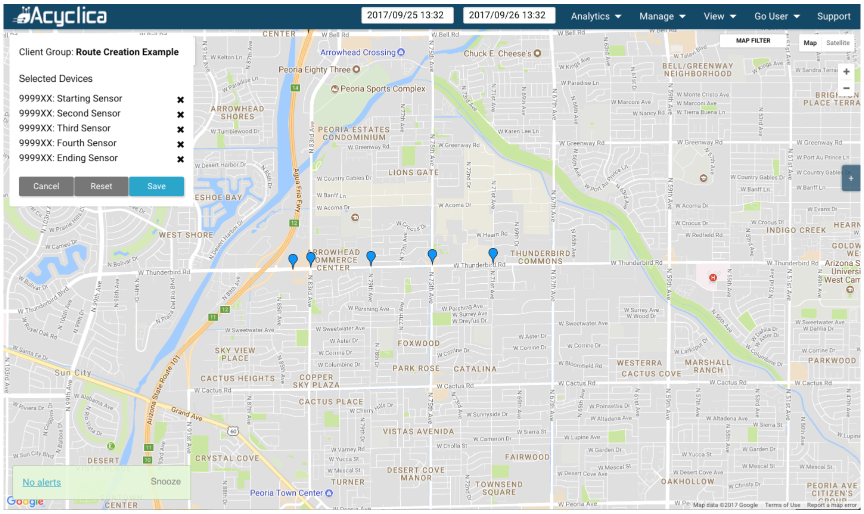

Figure 3.6 - Select all sensors in the order they appear in the route. As you do, adjacent sensors will be selectable

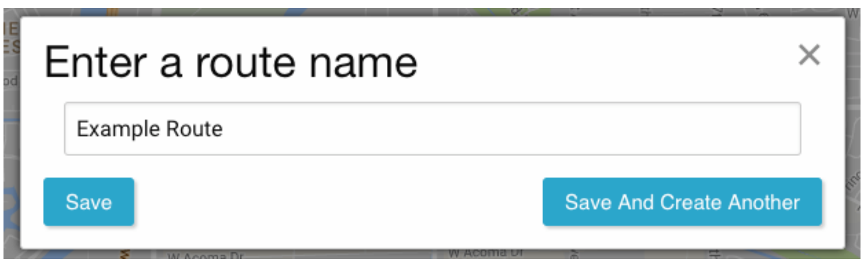

Figure 3.7 - Once the route is fully selected, click Save

1. Select the route.
2. In the sidebar, select *Manage Travel Time Alerts*.
3. Select the days of the week for the alert.
4. Select the hours of the day for the alert.
5. Enter a travel time threshold in seconds. Travel times over this threshold will bin together to potentially trigger the alert. (Defaults to 120 seconds)
6. Enter a number of times a travel time must be over the threshold before it triggers the alert. (Defaults to 3)
7. Enter the time frame for a travel time alert to be binned. (Defaults to 15 minutes)
8. Select *Save* to save this travel time alert.

3.3 Origin-Destination (OD) Groups

1. Hover over *Manage* in the *Top Navigation Bar* and select *Add OD Group*
2. Select the sensors that will be part of the group. This can be done in any order.
3. Select *Save*.
4. In the dialog box that appears, name your OD Group and select *Save*.

Note: The *Reset* button will clear all sensors from the selected sensor list, while the *Cancel* button will exit the OD Group tools without saving any changes.

3.4 Volume Speed and Occupancy

Volume Speed and Occupancy (VSO) sensors are available from Sensys and FLIR; in addition to providing those three metrics, they can also be combined with Travel Time and other route-based data for more advanced analytics.
Initial setup of a VSO sensor must be done by a member of the support team at Acyclica. If you wish to add any VSO sensors to your client group, please contact support@acyclica.com to do so. Please have the serial of the sensor or sensors you wish to associate with the VSO sensor(s).

Once a VSO sensor has been added to your client group, a configuration admin will be able to set the location and other configuration parameters.

1. Click the *Search Bar* and enter the serial number of the sensor that should be associated with the VSO sensor.
2. In the resulting list, locate the sensor name starting with the sensor type that represents your VSO sensor (eg. a FLIR camera associated with sensor 999000 would be listed as "FLIR – 999000:").
3. Open the *Hamburger Menu* (three horizontal lines), next to the sensor to be edited.
    - To modify the name, short name, and/or description of the VSO sensor, select *Edit Name and Description*
    - To modify the location of the sensor, select *Move Contents* and follow the procedure for moving a sensor, as listed in part 3 of Section 3.1.2
4. To associate a VSO sensor with an existing sensor, select *Move Contents* and then locate the map marker for the sensor sensor you want to connect, then select *Save*.

Section 4

Analytics

4.1 Algorithms

Acyclica uses five different algorithms to calculate travel time. These provide insight into different aspects of traffic conditions in order to develop a complete picture of roadway conditions. Figure 4.1: is an illustration of the five algorithms and the detection points that are used to calculate them. Travel time is calculated on a per-segment basis, and are summed to provide route travel time. By default, the Strength, Maximum and Minimum travel times will be displayed. Display of these can be toggled temporarily when running analytics, and defaults can be changed in Settings > Preferences.

- **Strength** – measures travel time from stop bar (Point B) to stop bar (Point B) between intersections. This is the best single representation of what a driver might expect to see when traveling a route
- **Minimum (Min)** – calculates travel time from the last detection (point C) at the first intersection, to the first detection (Point A) at the second intersection. This excludes delay at the intersections, and provides a representation of free flow travel time
- **Maximum (Max)** – calculate travel time from the first detection (Point A) at the first intersection to the last detection at the second intersection (Point C). By including all potential delay, this shows the maximum length of time it could take to traverse the route
- **First** – travel time is calculated from the first detection (Point A) at the first intersection to the first detection (Point A) at the second intersection. This includes delay from the first intersection while excluding it from the second. Looking at data this way will have the greatest effect when looking at routes that are made up of smaller numbers of sensors.
- **Last** – travel time is calculated from the last detection (Point C) at the first intersection to the last detection (Point C) at the second intersection. This excludes delay from the first intersection and excludes it from the second. Looking at data this way will have the greatest effect when looking at routes that are made up of smaller numbers of sensors.

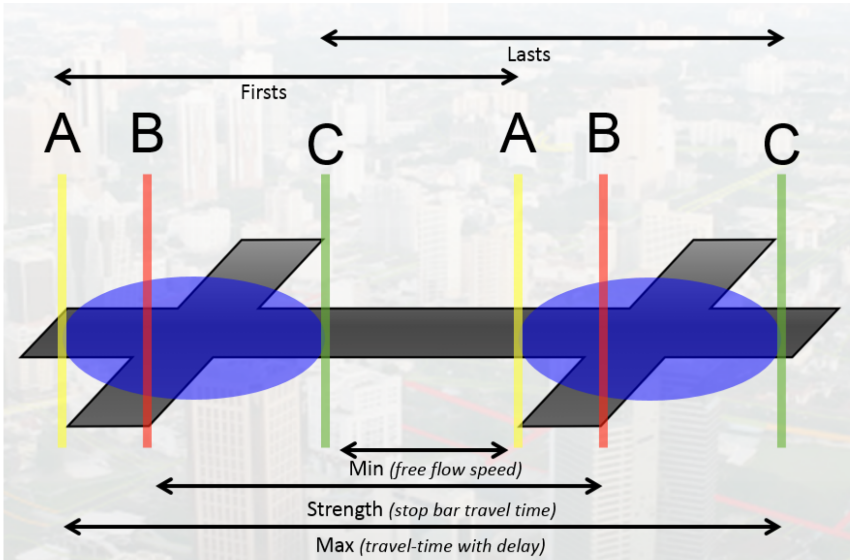

Figure 4.1 - The Five Travel Time Algorithms. Traffic flows from left to right

4.2 Running Analytics

All analytics are run using the same basic information.

- **Time Interval** – use the date/time picker to set the interval over which the analytic will be run. Some analytics have maximum intervals over which they can be run (e.g. 3 hours, 7 days, etc.). The maximum intervals are listed with each of the analytics in the descriptions that follow. Maximum intervals are based on the amount of data that needs to be loaded and the time that it can take to download and display the data. On slower connections it may be necessary to use shorter intervals in order to ensure that they can be displayed properly.
- **Data Source** – the Route, Sensor, or VSO Location for which the analytic will be run. These can be selected using Map Markers and the Information Panel, or the Search Field.
- **Analytic** – after the Time Interval and Data Source have been selected, the Analytic can be run from the Information Panel or the menu in Top Nav. Different Analytics will require different Data Sources. For example, Route-Based Analytics require a Route as the Data Source. If the correct Data Source has not been selected prior to running an Analytic, the system will prompt the selection of one. Analytics are organized by Data Source in the descriptions below.

4.3 Analytics Modules

Please note, Performance Metrics and Signal Timing Tools are optional added-cost modules and may not be available if they weren't included with the original purchase. If you have questions about whether these should be available.

4.4 Route Tools

4.4.1 Travel Time by Segment

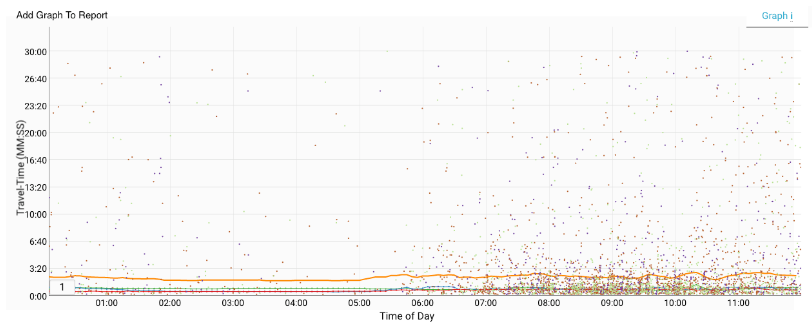

Figure 4.2 - Travel Time by Segment

Shows travel time for each Segment within a route, and for the Route as a whole. Travel times for segments make it easier to see whether congestion on a single segment, multiple segments, or entire route is responsible for elevated travel time. For intervals longer than 24 hours, hovering over the chart will display travel times for each of the segments and the route on the right side of the page. If the selected period of time is less than 24 hours, individual matches will be displayed as points on the graph, as shown in Figure 4.2, and hovering over the chart will show travel times for individual matches. This analytic has a maximum time frame of 4 months or 124 days.

4.4.2 Travel Time

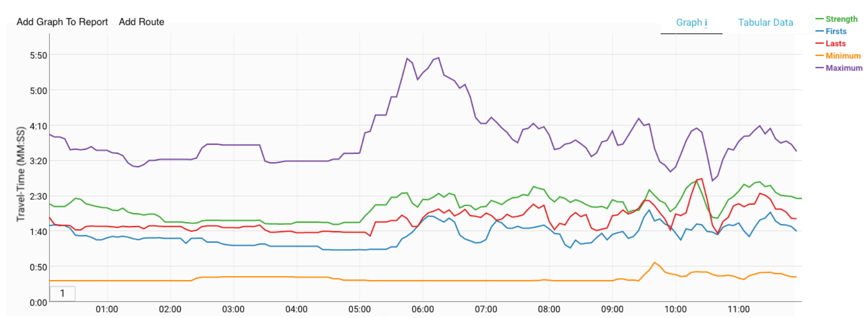

Figure 4.3 - Travel Time

Displays travel time for a route. The five travel time algorithms provide insight into different aspects of travel time, including - typical travel time (strength), free flow without delay (min), and all potential delay (max), as shown in Figure 4.3. Hovering over the chart will show the travel time based on each of the selected algorithms. This analytic has a maximum time frame of 4 months or 124 days.

4.4.3 Speed

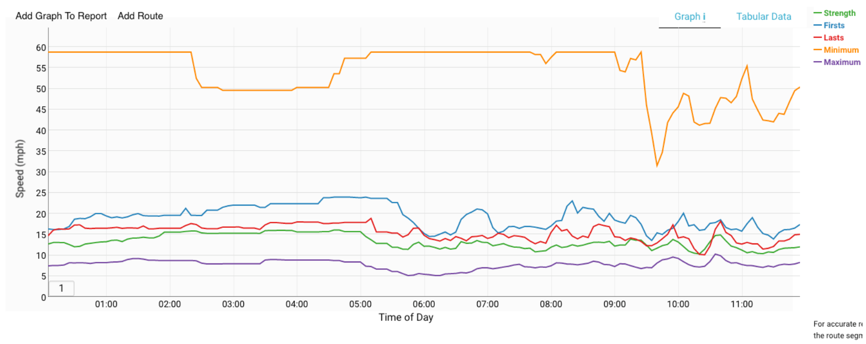

Figure 4.4 - Speed

Calculates speed along a route using route length and the five travel time algorithms. Because speed is calculated using travel time and route length, it will provide average speed along the route and not instantaneous speed. In order for speeds to be calculated accurately, it is necessary for all segments within the route to have overlays. as shown in Figure 4.4. Hovering over the chart will show the time the cursor is over, and the average speed for the route at that time. This analytic has a maximum time frame of 4 months or 124 days.

4.5 Signal Timing Tools

4.5.1 Route

4.5.1.1 Intersection Route Delay

Displays the average length of time that WiFi devices spent at each intersection along a route, as shown in Figure 4.5. This can be helpful seeing the relative delay at intersections that make up a route, making it easier to see which ones are performing worst overall. Hovering over each of the columns will display the name of the intersection (sensor) and its delay on the right side of the page. This analytic has a maximum time frame of 4 months or 124 days.

4.5.1.2 Progression Diagram

Plots speed as a function of distance and travel time for each segment along a route. The X axis displays time in minutes and seconds, and the Y axis displays distance in miles (or KM if selected in User Preferences). Steeper slopes indicate higher speeds. The Progression Diagram provides a visual display of how segments within a route are performing relative to each other. Segments where the slope flattens are performing worse than those around them, making it easy to see where performance is

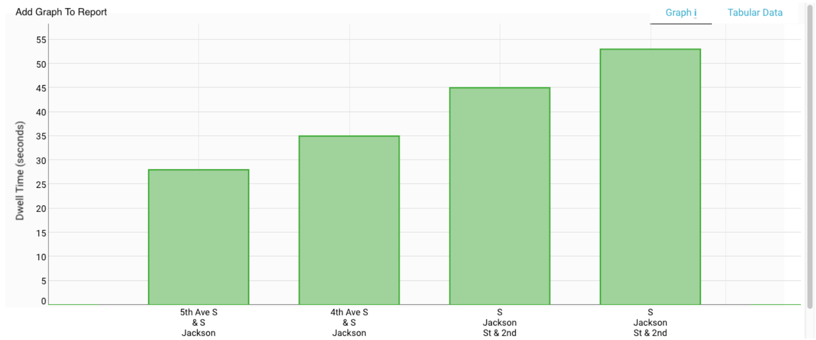

Figure 4.5 - Intersection Route Delay

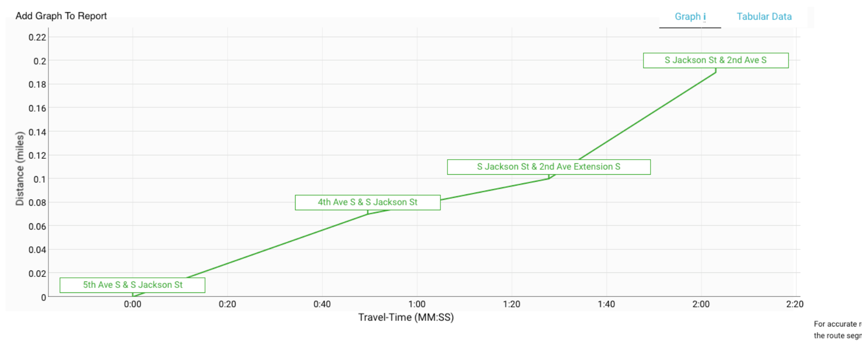

Figure 4.6 - Progression Diagram

not what it should be, as shown in Figure 4.6. Hovering over the chart will display information about distance and travel time. This analytic has a maximum time frame of 4 months or 124 days.

4.5.1.3 Speed by Segment

Similar to Travel Time by Segment, this analytic displays the average speed of vehicles traveling each of the segments of a route as well as average speeds for the route as a whole, as shown in Figure 4.7. Seeing speed for each of the segments makes it easier to see which may be responsible for performance problems. This analytic has a maximum time frame of 4 months or 124 days.

4.5.1.4 Timing Plan Analysis

Displays travel time for user-defined intervals for each day of a week, for a single week. as shown in Figure 4.8. To look at a single interval during two different weeks, use the Comparison option within Timing Plan Analysis. This analytic has a maximum time frame of one week or seven days.
When running Timing Plan Analysis, the *Name* and *Client Group* fields need only be completed if the

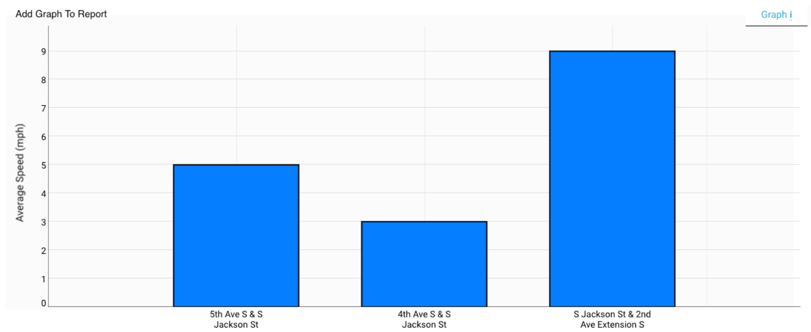

Figure 4.7 - Speed by Segment

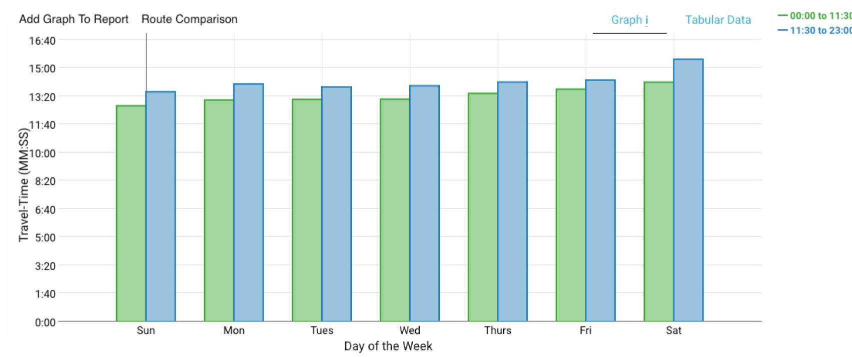

Figure 4.8 - Timing Plan Analysis

plan is to be saved. Otherwise, the Timing Plan to be analyzed may be entered in the Start and End Time, adding more rows as necessary with the plus (**+**) button. Once this plan has been constructed, select **Run** (or **Save & Run**, to save a custom plan) and wait for the analytic to load. If a timing plan has been previously created, it can be found under the **Saved** section, from where it may be selected and run.

4.5.1.5 Travel Time by Day of Week

Provides a representation of typical travel conditions for each day of the week. Travel Times for each instance of each day of the week during the selected interval are averaged together in the intervals selected in Data Resolution. For example, if 5 minutes is selected for the data resolution, the data from every five minute interval in every Tuesday during the selected are averaged to create a typical Tuesday, and so forth. Figure 4.9. This analytic has a maximum time frame of 4 months or 124 days. Selecting longer intervals will decrease the influence of anomalous travel conditions on individual days. Hovering over the chart will show the time of day, and travel time in minutes and seconds for each day of the week.

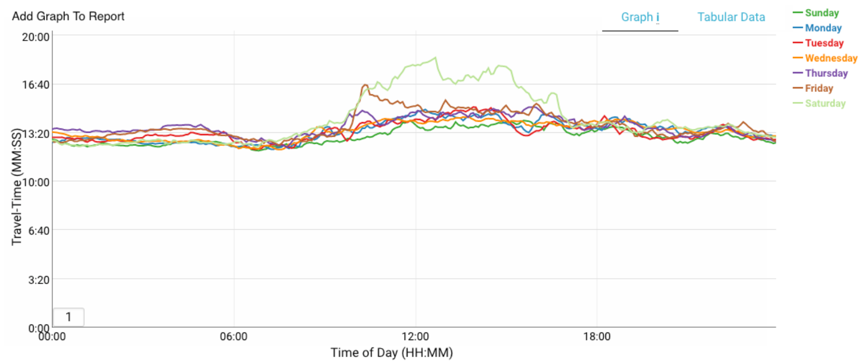

Figure 4.9 - Travel Time by Day of Week

4.5.1.6 Week Over Week Travel Times

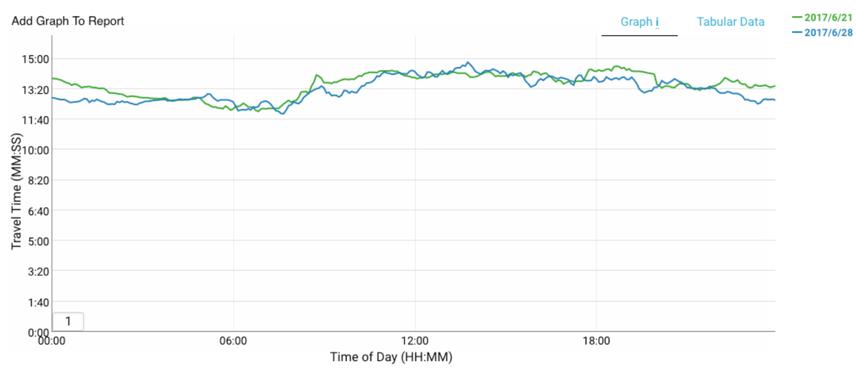

Figure 4.10 - Week Over Week Travel Times

This provides a different type of comparative analysis for days of week. Instead of showing what a typical day of the week looks like, it will show travel time for every instance of a specific day of the week within a selected time frame. For the chosen day of the week, travel times for every week within the selected interval of time will be displayed. Each is given a separate line within the graph to provide easy comparisons, as shown in Figure 4.10. Some places where this type of comparison may be useful are when there is an event that you'd like to compare to more typical travel conditions, or before and after comparison when making timing changes. Hovering over the chart will show the time of day, date and travel time in minutes and seconds. This analytic has a maximum time frame of 4 months or 124 days.

4.5.1.7 Timing Run

Provides data similar to what can be provided by a float car study. Each timing run is a single WiFi sensor that was seen at every sensor along a route. Devices that miss one or more sensors aren't

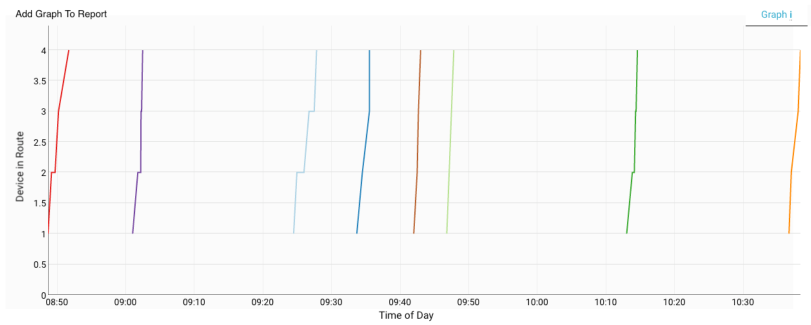

Figure 4.11 - Timing Run

included, as shown in Figure 4.11. Each timing run displayed includes the length of time taken to travel between sensors as well as delay incurred at intersections along the route. Hovering over the chart will show the number of the timing run, and the number of the sensor at which the data was collected. Because individual devices must be seen at every sensor along a route, this analytic works best on higher volume routes with lower numbers of sensors. Longer routes with lower volume will have smaller numbers of vehicles to display. This analytic has a maximum time frame of 1 day.

4.5.2 Device

4.5.2.1 Delay by Movement

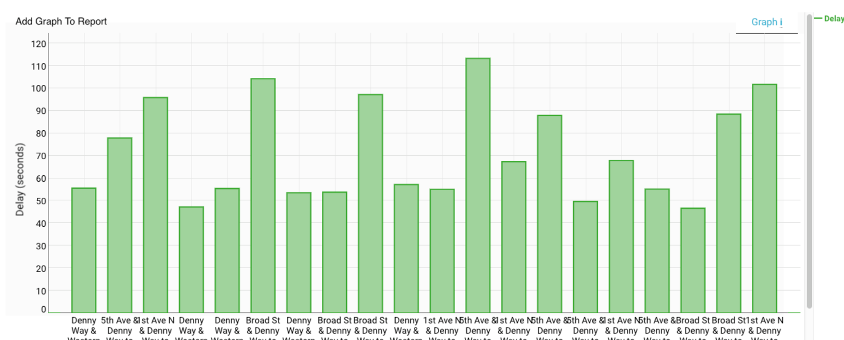

Figure 4.12 - Delay by Movement

On a per-approach basis, the average time WiFi sensors are seen at an intersection, as shown in Figure 4.12. The labels for movements can be thought of as origin and destination pairs, with the selected intersection as a mid-point. For each movement, the column label will include the name of the origin sensor that a device was seen at prior to going through the selected intersection, and the destination sensor that the device passed through after going through the selected intersection. Hovering over a column shows delay in seconds. This analytic has a maximum time frame of 1 month or 30 days.

4.5.2.2 Delay by Approach

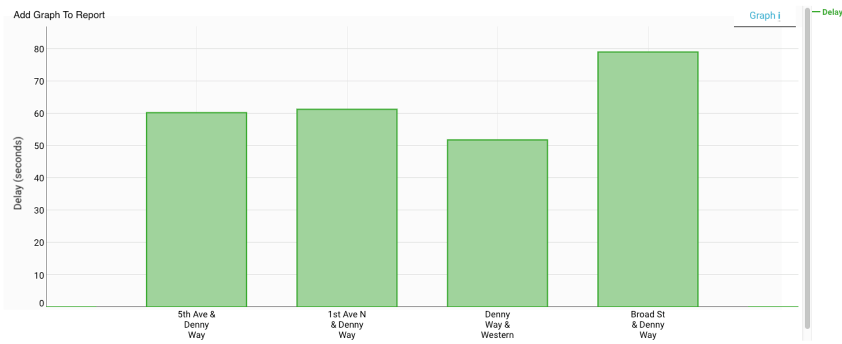

Figure 4.13 - Delay by Approach

On a per-approach basis, the average time WiFi sensors are seen at an intersection, as shown in Figure 4.13. The labels for approaches are the name of the last sensor that a device was seen at before going through the selected intersection. Hovering over the columns will show delay in seconds to the right of the chart. This analytic has a maximum time frame of 1 month or 30 days.

4.6 Performance Metrics

4.6.1 Route

4.6.1.1 Cumulative Route Delay

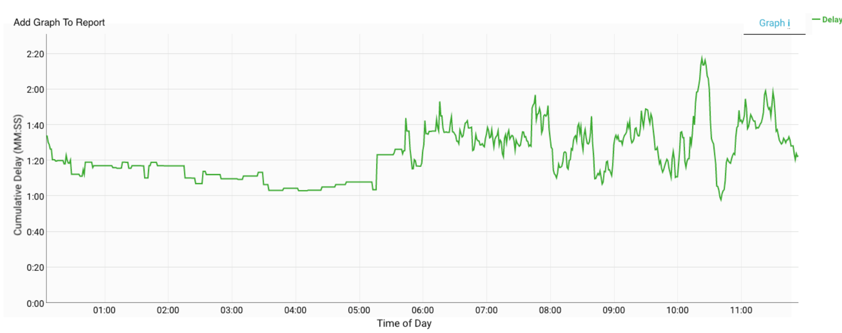

Figure 4.14 - Cumulative Route Delay

The cumulative delay that individual vehicles traveling a route encounter. The delay incurred at each intersection is summed to find a total, as shown in Figure 4.14. Hovering over the chart will show the date and time of day, and delay in minutes and seconds. This analytic has a maximum time frame of 4 months or 124 days.

4.6.1.2 Congestion Index

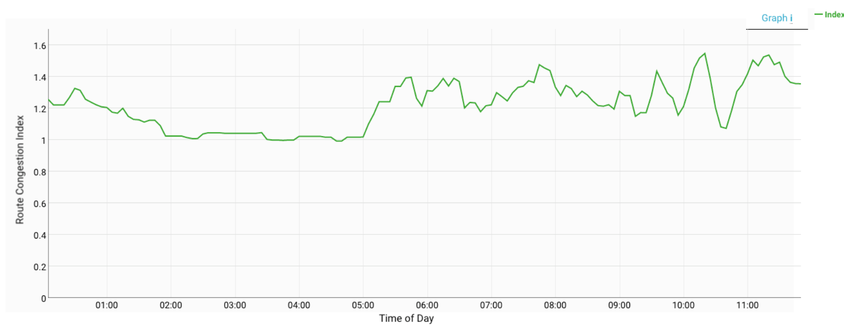

Figure 4.15 - Congestion Index

Shows travel-time as a multiplier of the estimated free-flow travel-time 1.0 represents free-flow conditions and higher numbers such as 1.25 indicate more congestion, as shown in Figure 4.15. This analytic has a maximum time frame of 4 months or 124 days.

4.6.1.3 Congestion Overlay

Shows whether current conditions are better or worse than average. The current fifteen minute interval is compared to average travel times from the previous two weeks during the same interval of time on the same day of the week.Hovering over the chart will show the date and time, and the multiplier of that time's travel times compared to the historical data. 1.2 would mean that current travel times are 120 percent of the historical data. This analytic has a maximum time frame of 4 months or 124 days.

4.6.1.4 Vehicle Hours Traveled

Uses volume data from count stations and travel time data to calculate the total number of vehicle hours traveled on a route during the selected interval of time. Calculated by multiplying counts by the average travel time, as shown in Figure 4.16. Hovering over the chart displays the number of vehicle hours traveled in five minute increments on the right side of the page. This analytic has a maximum time frame of 4 months or 124 days.

4.6.1.5 Vehicle Miles Traveled

Uses count data and route distance to calculate the total number of vehicle miles that were traveled along the selected route during the chosen period of time. Vehicle Miles Traveled are found by multi-

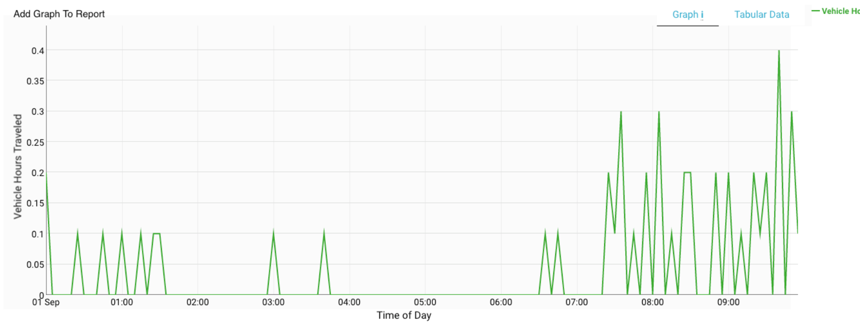

Figure 4.16 - Vehicle Hours Traveled

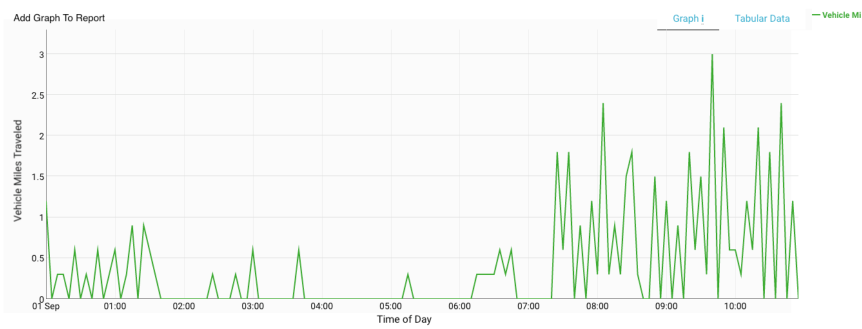

Figure 4.17 - Vehicle Miles Traveled

plying counts by route length, as shown in Figure 4.17. Hovering over the chart displays the number of vehicle miles (or kilometers, depending on settings) traveled on the right side of the chart. This analytic has a maximum time frame of 4 months or 124 days.

4.6.1.6 VMT vs VHT

Provides a scatter plot based on Vehicle Miles Traveled and Vehicle Hours Traveled, as shown in Figure 4.18. This analytic has a maximum time frame of 4 months or 124 days.

4.6.1.7 Idle Emissions

Uses vehicle counts and travel times to calculate emissions attributable to idling. EPA estimates for idle emissions are multiplied by the difference between current conditions and free flow on a per-vehicle basis to find emissions. This analytic has a maximum time frame of 1 month or 31 days.

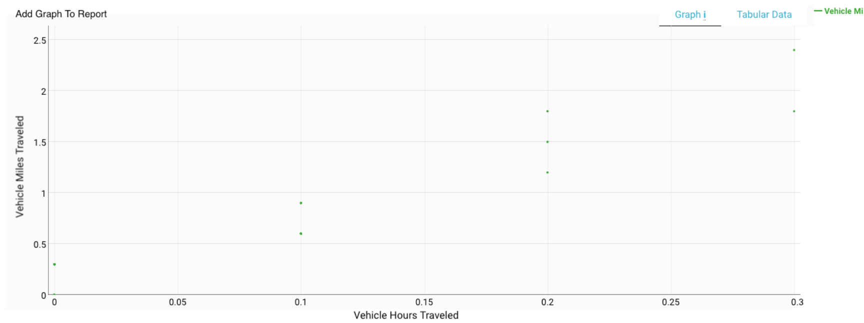

Figure 4.18 - VMT vs VHT

4.6.1.8 Vehicle Delay Hours

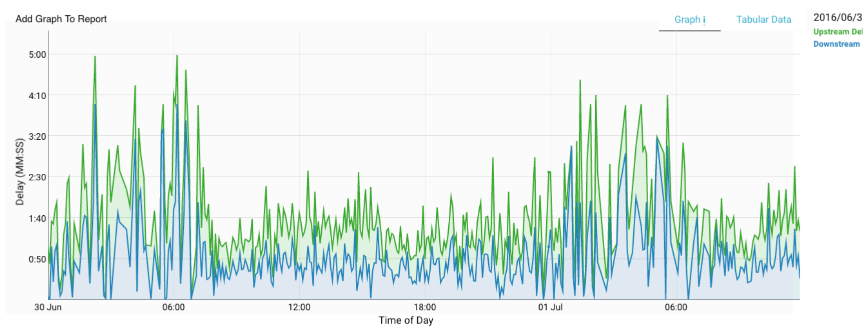

Figure 4.19 - Vehicle Delay Hours

Hours of vehicle travel times that are attributable to delay. It is found by multiplying delay by vehicle counts.

4.6.1.9 Congestion Emissions

Multiplies travel time due to congestion with emissions averages to find emissions due to congestion. Emissions averages used in calculations are sourced from the EPA. This analytic has a maximum time frame of 1 month or 31 days.

4.6.2 Cabinet

4.6.2.1 Arrivals on Red / Green

Shows a breakdown of when cars arrive at the intersection. Arrivals during green will contribute to that percentage. Arrivals on yellow and red will count toward red. This analytic requires that Activation Detectors and RYG Channels be selected for each phase for which the analytic will be run. This analytic has a maximum time frame of 1 week or 7 days.

4.6.3 Cabinet

4.6.3.1 Purdue Coordination Diagram

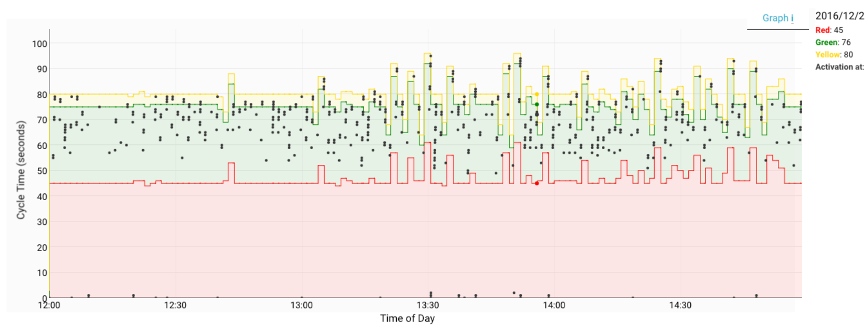

Figure 4.20 - Purdue Coordination Diagram

Shows where activations occur during the phase cycle to provide a visual representation of when platoons of cars are arriving, as shown in Figure 4.20. Ideally, platoons should arrive on green toward the beginning of phase. Platoons arriving later or in other parts of the phase cycle indicate that intersection timing could be adjusted to provide better performance. The phase can be selected on the right side. This analytic requires that Activation Detectors and RYG Channels be selected for each phase for which the analytic will be run. This analytic has a maximum time frame of 1 week or 7 days.

4.6.4 Cabinet

4.6.4.1 Pedestrian Delay

Calculates the length of time from Ped button activation to service, showing how long pedestrians wait until they are given a walk sign. The dropdown on the left side of the chart can be used to select the phase for which data will be displayed. Hovering over the chart will show the date and time, and Pedestrian Delay in seconds. This analytic has a maximum time frame of 1 week or 7 days.

4.6.5 Cabinet

4.6.5.1 Vehicle Delay

Calculates the length of time from detector activation until green phase. Times are provided for average vehicle delay and for total vehicle delay. Hovering over the chart will display the date and time interval, and the Average Vehicle Delay and Total Vehicle delay in HH:MM:SS format. The phase can be selected using the dropdown to the left of the chart. This analytic has a maximum time frame of 1 week or 7 days.

4.6.6 Device

4.6.6.1 Arrival and Departure Delay

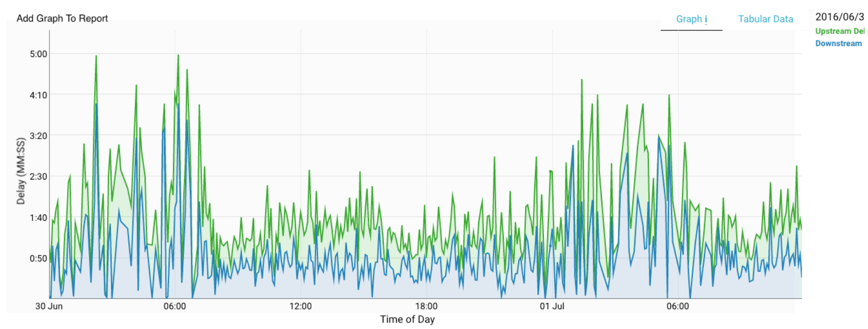

Figure 4.21 - Arrival and Departure Delay

Illustrates whether delay is incurred when approaching an intersection, or after passing through, as shown in Figure 4.21. Hovering over the chart will show the date and time, and the delay in minutes and seconds that occurred before (Upstream) or after (Downstream) vehicles have passed through the intersection. This analytic has a maximum time frame of 1 month or 31 days.

4.7 Device Tools

4.7.1 Data Availability

Each square represents a single calendar day. The color of the square indicates the volume of data collected by the selected sensor on that day. Darker blue indicates a larger volume of data collected by the sensor. Hovering over each square provides the number of records collected, as shown in Figure 4.22. This analytic has a maximum time frame of 12 months or 365 days.

4.7.2 Records per Hour

Displays two things - the number of unique WiFi devices seen by a sensor, and the total number of probe requests seen from all WiFi devices, as shown in Figure 4.23. Hovering over the chart will dis-

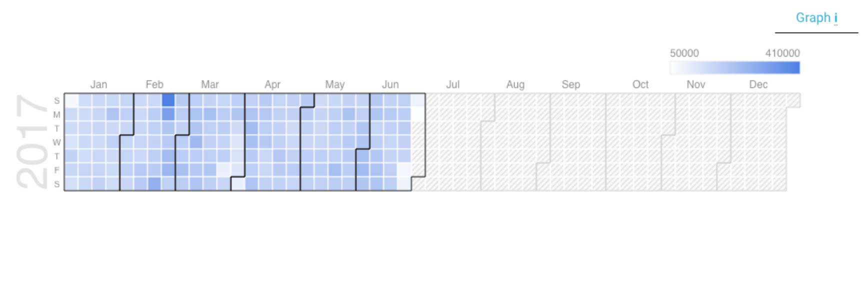

Figure 4.22 - Data Availability

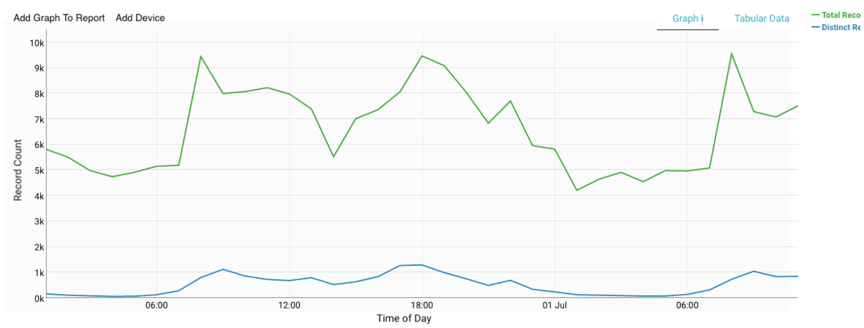

Figure 4.23 - Records per Hour

play the date and time interval, as well as the number of distinct records, and the total number of records. This analytic is useful when looking at a sensor to determine when it has been online and offline, such as when there have been issues with network connectivity. It can also be used to ensure that adequate data is being collected to calculate travel time. This analytic has a maximum time frame of 12 months or 365 days.

4.7.3 Distinct Records per Day

The number of unique WiFi devices seen at a sensor per day. Each WiFi device is counted only once, as shown in Figure 4.24. Hovering over the bars will display the date and the number of unique WiFi devices seen that day. This analytic has a maximum time frame of 12 months or 365 days.

4.7.4 Delay

The average time WiFi sensors spend around an intersection. Average time is displayed as a line. The length of time each sensor was seen is also plotted as a point, as shown in Figure 4.25. Hovering over the chart will show the delay for individual WiFi devices, and the overall average delay at that time. This analytic has a maximum time frame of 1 month or 31 days.

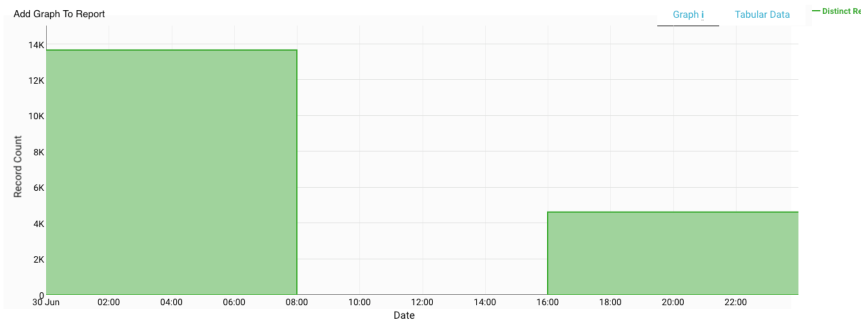

Figure 4.24 - Distinct Records per Day

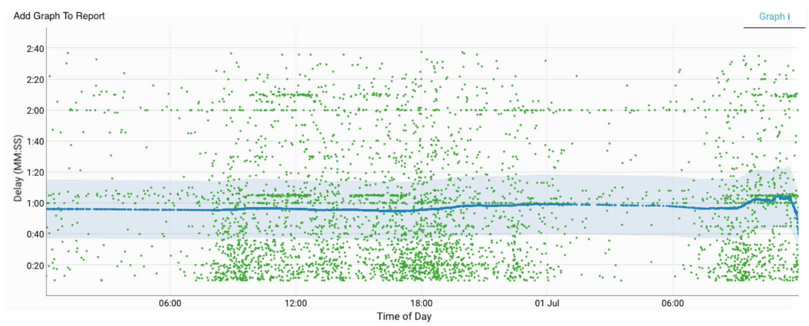

Figure 4.25 - Delay

4.8 VSO Analytics

4.8.1 Data Availability

A visual representation of the volume of data collected by VSO sensors. Darker blue indicates that a larger number of records were recorded, as shown in Figure 4.26. Hovering over the chart shows the volume collected. This analytic has a maximum time frame of 1 year or 365 days.

4.8.2 Time of Day Distribution

The distribution of overall traffic volume collected during the following parts of the day: Early AM (before 7:00 am), AM Peak (7:00-9:00 am), Midday (9:00 am - 4:00 pm), PM Peak (4:00-6:00 pm), and Evening (after 6:00 pm), as shown in Figure 4.27. This analytic has a maximum time frame of 4 months or 120 days.

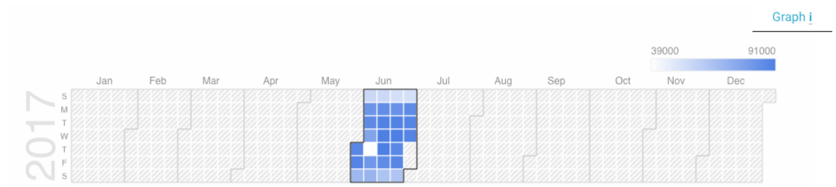

Figure 4.26 - Data Availability

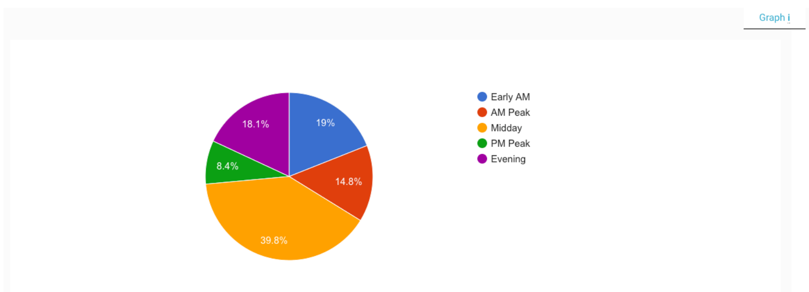

Figure 4.27 - Time of Day Distribution

4.8.3 Volume

Number of vehicles counted per minute, per detector, over the selected time period, as shown in Figure 4.28. Options are available to look at counts overall, by individual count station, or by count zones. Hovering over the chart will show the date and time, and volume. If more than one count station, or zone is displayed, these will be labeled in the display on the right side. This analytic has a maximum time frame of 4 months or 120 days.

4.8.4 Speed

Average speed of vehicles on a per detector basis, updated every minute, as shown in Figure 4.29. These are instantaneous speeds as measured by the detector and not average speeds along segments or routes. Options are available to look at speed overall, by individual count station, or by count zones. Hovering over the chart will show the date and time, and speed. If more than one count station, or zone is displayed, these will be labeled in the display on the right side. This analytic has a maximum time frame of 4 months or 120 days.

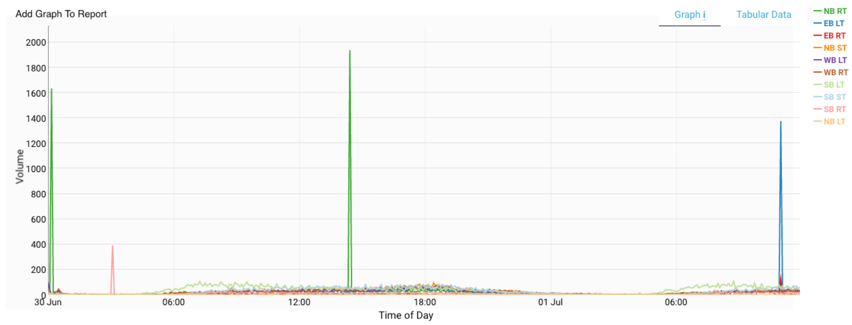

Figure 4.28 - Volume

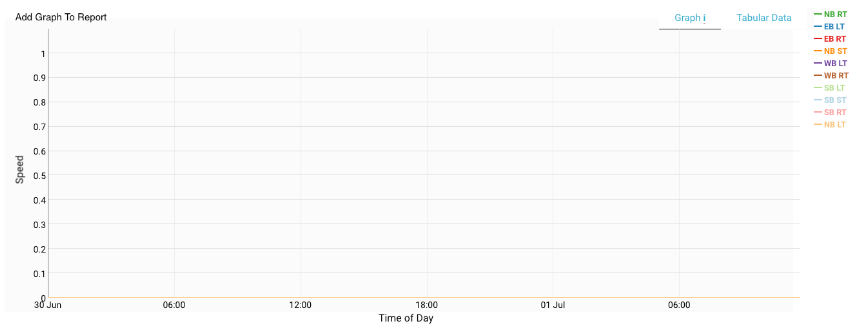

Figure 4.29 - Speed

4.8.5 Occupancy

Percent of time that the detection zone of a detector is occupied by a vehicle, as shown in Figure 4.30. Options are available to look at occupancy overall, by individual count station, or by count zones. Hovering over the chart will show the date and time, and occupancy. If more than one count station, or zone is displayed, these will be labeled in the display on the right side. This analytic has a maximum time frame of 4 months or 120 days.

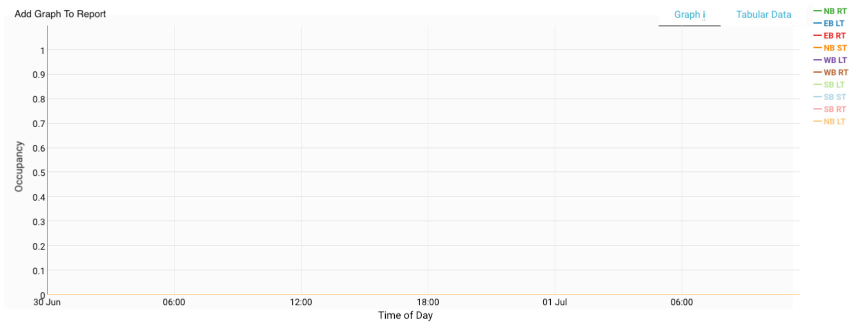

Figure 4.30 - Occupancy

4.9 Origin / Destination

4.9.1 OD Groups

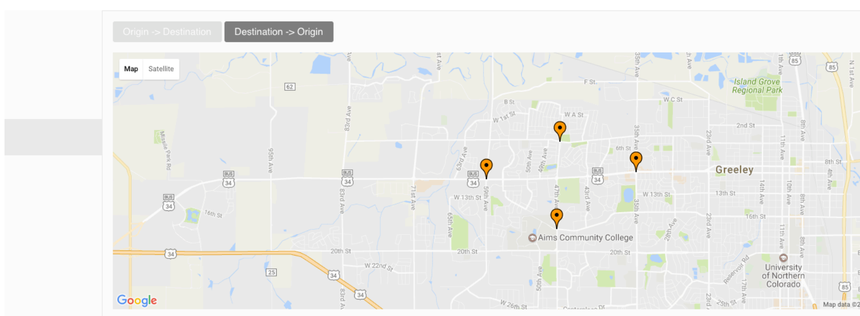

Figure 4.31 - OD Groups

A flexible Origin-Destination Trip Table. The system calculates traversals between the sensors included in the O-D group. Any sensor can be set as an origin or destination. Trips can be displayed as percentages or as counts of traversals. Travel time for each of the segments is also calculated. The Map view is an interactive display where sensors can be selected and percentages of trips originating at, or going to other sensors are displayed, as shown in Figure 4.31. This analytic has a maximum time frame of 1 month or 31 days.

Appendix A

Alerts

A.1 Access Alerts

A.1.1 To access the Alerts Creation Tool:

1. Login to Go.

2. Hover over **Manage** and select **Create Alert**.

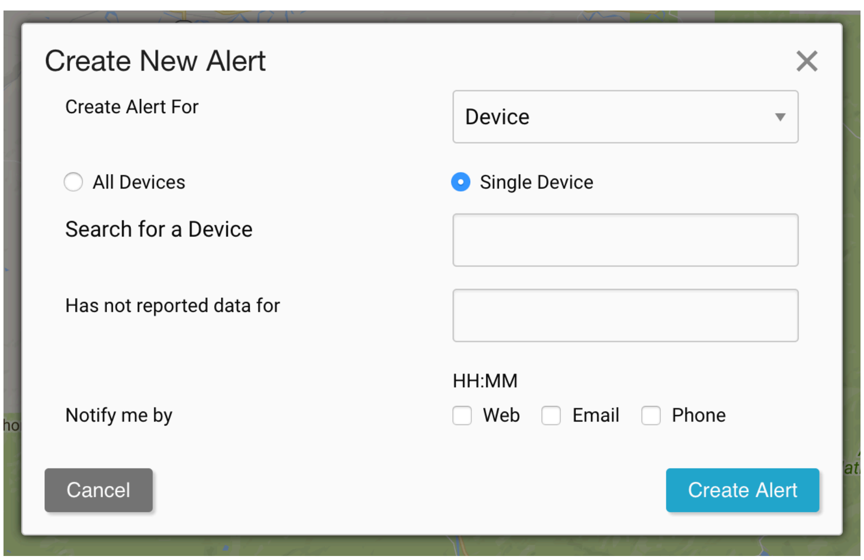

Figure A.1 - The Alerts Creation Tool modal.

A.2 Alert Creation

A.2.1 To create an Alert for a Device:

1. Select **Device** in the Alert Creation Tool.
2. Choose whether you want to create an Alert for a single Device or for all Devices in a Client Group.
   a) If *Single Device* is selected, search for the Device's serial.
   b) If *All Devices* is selected, the Alert will be assigned to all Devices in the selected Client Group.
3. Enter the threshold at which the alert will be triggered formatted as *HH:MM*.
   a) The minimum threshold for an Alert is 10 minutes.
   b) The maximum threshold for an Alert is 24 hours.
   c) In areas with known connectivity issues, a longer interval may be preferred to avoid false positives.
   d) Where it is important to have uninterrupted travel times, such as where information is being displayed on variable message signs, a shorter interval may be preferred.
4. Select the notification methods for the alert. If the alert is triggered, all selected methods will be used to contact you.

A.2.2 To create and Alert for a Traffic Cabinet:

1. Select **Cabinet** in the Alert Creation Tool.
2. Choose whether you want to create an Alert for a single Traffic Cabinet or for all Traffic Cabinets in a Client Group.
   a) If *Single Cabinet* is selected, search for the Cabinet's serial.
   b) If *All Cabinets* is selected, the Alert will be assigned to all Traffic Cabinets in the selected Client Group.
3. User the drop down, select the type of Cabinet Alert you wish to create.
4. Select the notification methods for the alert. If the alert is triggered, all selected methods will be used to contact you.

A.2.3 Alert Methods

- **Web** — A web alert will appear in the Alerts Notification area, in the bottom left of the web interface.
- **Email** — An Alert will be sent to the email address linked to your account.
- **Phone** — An SMS containing the Alert will be sent to the phone number set in your *Account Settings* page. At this time, if you do not have a phone number set, this option will not alert you.

A.3 Alert Management

A.3.1 To access the Alert Management Window:

1. Login to Go.
2. Hover over **Manage** and select **All Alerts**.
3. The Alerts Management Window will open.

A.3.2 To Manage Alert Subscriptions:

1. Find the Alert or Alerts to which you wish subscribe. The search field can be used to quickly find an Alert by type or Device/Traffic Cabinet serial number.
2. Select the type of Alert you wish to receive using the check boxes.
3. Select **Save** to update your Alert preferences.

Note: At this time, when you **Delete** an Alert, it retains the Alert setup, but unsubscribes you, and any other user using that Alert, from the selected Alert.

Appendix B

API

B.1 About

All available API methods are HTTP GETs.

Please contact support@acyclica.com to get your API key and to learn more about the limits on our API.

Successful calls will correctly include HTTP Status Code 200, while failures will include appropriate 4XX and 5XX Codes.
The default endpoint host is https://go.acyclica.com.

B.2 Sensor

B.2.1 Sensor Inventory

B.2.1.1 About

This provides an XML-formatted listing of all of the sensors that your user has the ability to access, with the following information about each sensor:

- Sensor Serial
- Sensor Name (Long form, as set in Acyclica Go)
- Sensor Description (As set in Acyclica Go)
- Latitude
- Longitude
- LastReport (the last time the sensor reported data)

B.2.1.2 Endpoint

/datastream/device/inventory/xml/[API_KEY]

B.2.1.3 Parameters

- **[API_KEY]**
  - A generated user access token used to authenticate a user against the Acyclica Go platform. Contact support@acyclica.com to receive your API key.

B.2.1.4 Example Output

XML

```xml
<AcyclicaSensorInventory>
    <AcyclicaSensor>
        <Serial>Sensor Serial Number</Serial>
        <Name>Sensor Name</Name>
        <Description>Sensor Description</Description>
        <Latitude>39.7537229580521</Latitude>
        <Longitude>-105.001809597015</Longitude>
        <LastReport>1489588555</LastReport>
    </AcyclicaSensor>
    ...
</AcyclicaSensorInventory>
```

B.2.2 Sensor Data

B.2.2.1 About

This provides the data from one requested sensor over the time frame requested.

B.2.2.2 Endpoint

/datastream/device/[FORMAT]/time/[API_KEY]/[DEVICE_ID]/[START_TIME]/[END_TIME]

B.2.2.3 Parameters

- **[FORMAT]**
  The format of the data you are querying, either ”xml” or ”csv”
- **[API_KEY]**
  - A generated user access token used to authenticate a user against the Acyclica Go software. Contact support@acyclica.com to receive your API KEY.

- **[DEVICE_ID]**
  - The 4–7 digit number that represents an individual diffrf sensor, cabinet, or other sensor.
- **[START_TIME]**
  - The beginning time in the range you are searching. Time values must be in epoch format, in **milliseconds**. March 15, 2017 00:00 would be represented by `1489536000000`.
- **[END_TIME]**
  - The end time in the range you are searching. Time values must be in epoch format, in **milliseconds**. March 15, 2017 03:00 would be represented by `1489546800000`.

B.2.2.4 Example Output

XML

```
<TrafficSensor ID="" Owner="">
  <HardwareType/>
  <FirmwareType>Independent</FirmwareType>
  <Group/>
  <Detector ID="">
      <HardwareType>Wireless Probe</HardwareType>
      <Name>[Sensor Name]</Name>
      <Description>[Sensor Description]</Description>
      <Data>
          <Time format="unix" zone="utc" capture="first">1489104000.898</Time>
          <Hash>
              [Match_1]
          </Hash>
          <Strength>-51</Strength>
          <SensorType>[Sensor_ID]</SensorType>
      </Data>
      <Data>
          <Time format="unix" zone="utc" capture="first">1489104000.898</Time>
          <Hash>
              [Match_2]
          </Hash>
          <Strength>-50</Strength>
          <SensorType>[Sensor_ID]</SensorType>
      </Data>
      ...
  </Detector>
</TrafficSensor>
```

CSV

```
Timestamp    MAC Hash    Strength    Serial
1489536000.112  [Match_1]    -44 999912
1489536000.151  [Match_2]    -41 999912
1489536000.197  [Match_3]    -50 999912
...
```

B.2.3 Sensor Record Count

B.2.3.1 About

This provides a JSON-formatted count of data records between two given times for a given sensor.

B.2.3.2 Endpoint

`/datastream/device/json/recordcount/[API_KEY]/[DEVICE_ID]/[START_TIME]/[END_TIME]`

B.2.3.3 Parameters

- **[API_KEY]**
  - A generated user access token used to authenticate a user against the Acyclica Go software. Contact support@acyclica.com to receive your API KEY.
- **[DEVICE_ID]**
  - The 4–7 digit number that represents an individual diffrf sensor, cabinet, or other sensor.
- **[START_TIME]**
  - The beginning time in the range you are searching. Time values must be in epoch format, in **milliseconds**. March 15, 2017 00:00 would be represented by `1489536000000`.
- **[END_TIME]**
  - The end time in the range you are searching. Time values must be in epoch format, in **milliseconds**. March 15, 2017 03:00 would be represented by `1489546800000`.

B.2.3.4 Example Output

JSON

```
{
    "serial": [
        "[SERIAL]"
    ],
    "count": 72810,
```

```
    "start_time": "2017-03-15 00:00:00 ",
    "end_time": "2017-03-15 10:00:00 "
}
```

B.2.4 Sensor Delay Matches

B.2.4.1 About

This provides an XML-formatted listing of hashed MAC addresses during the given time frame, which is limited to **1 minute**, as well as other information about this delay data. Times are in **seconds**.

B.2.4.2 Endpoint

`/datastream/device/delay/xml/time/[API_KEY]/[DEVICE_SERIAL]/[START_TIME]/[END_TIME]`

B.2.4.3 Parameters

- **[API_KEY]**
  - A generated user access token used to authenticate a user against the Acyclica Go software. Contact support@acyclica.com to receive your API KEY.
- **[DEVICE_ID]**
  - The 4–7 digit number that represents an individual diffrf sensor, cabinet, or other sensor.
- **[START_TIME]**
  - The beginning time in the range you are searching. Time values must be in epoch format, in **milliseconds**. March 15, 2017 00:00 would be represented by `1489536000000`.
- **[END_TIME]**
  - The end time in the range you are searching. Time values must be in epoch format, in **milliseconds**. March 15, 2017 03:00 would be represented by `1489546800000`.

B.2.4.4 Example Output

XML

```
<device id="">
    <name>Sensor Name</name>
    <description/>
    <delaydata>
        <data>
            <hash>
                Hashed MAC Address
```

```
            </hash>
            <first_time>1467266416.05</first_time>
            <strength_time>1467266416.05</strength_time>
            <last_time>1467266416.05</last_time>
            <timespan>0.0</timespan>
            <strongest>-63</strongest>
        </data>
        ...
    </delaydata>
</device>
```


B.3 Cabinet

B.3.1 Cabinet Inventory Listing

B.3.1.1 About

This call enumerates all of the cabinet locations available to your user, along with mapped RYG channels, if they are configured.

B.3.1.2 Endpoint

`/datastream/cabinet/meta/json/[API_KEY]`

B.3.1.3 Parameters

- **[API_KEY]**
  - A generated user access token used to authenticate a user against the Acyclica Go software. Contact support@acyclica.com to receive your API KEY.

B.3.1.4 Example Output

JSON


```
[
    {
        "longitude": -104.999303833,
        "id": CABINET_ID1,
        "configured_channels": [
            {
                "phase": 1,
                "name": "NB LT",
                "movement": 1,
```

```
                "approach": 1
            },
            {
                "phase": 3,
                "name": "NB RT",
                "movement": 1,
                "approach": 1
            },
            {
                "phase": 2,
                "name": "NB Thru",
                "movement": 1,
                "approach": 1
            },
            ...
        ],
        "latitude": 39.7512958333,
        "name": "Cabinet Name",
        "phases": [
            {
                "phase": 1,
                "name": "NB LT",
                "movement": 1,
                "approach": 1
            },
            {
                "phase": 3,
                "name": "NB RT",
                "movement": 1,
                "approach": 1
            },
            {
                "phase": 2,
                "name": "NB Thru",
                "movement": 1,
                "approach": 1
            },
            ...
        ]
    },
    {
        "longitude": -104.999303833,
        "id": CABINET_ID2,
        "configured_channels": [],
        "latitude": 39.7512958333,
```

```
        "name": "Cabinet Name 2",
        "phases": []
    },
    ...
]
```


B.3.2 Cabinet RYG Data

B.3.2.1 About

This will output the raw Red-Yellow-Green state data during the range of time requested, for the cabinet requested.

B.3.2.2 Endpoint

`/datastream/da/ryg/raw/[FORMAT]/[API_KEY]/[DEVICE_ID]/[START_TIME]/[END_TIME]`

B.3.2.3 Parameters

B.3.2.4 Example Output

JSON


```
[
    [
        1499644812000,
        6,
        3,
        1,
        1,
        2,
        1,
        1,
        1,
        3,
        1,
        1,
        1,
        0,
        1,
        3,
        1,
        0,
        0,
        0,
```

```
        0,
        0,
        0,
        0,
        0
    ],
    ...
]
```


XML


```
<data_aggregator_ryg_data>
    <data_aggregator_ryg_point>
        <timestamp>1499644812000</timestamp>
        <channel_1>6</channel_1>
        <channel_2>3</channel_2>
        <channel_3>1</channel_3>
        <channel_4>1</channel_4>
        <channel_5>2</channel_5>
        <channel_6>1</channel_6>
        <channel_7>1</channel_7>
        <channel_8>1</channel_8>
        <channel_9>3</channel_9>
        <channel_10>1</channel_10>
        <channel_11>1</channel_11>
        <channel_12>1</channel_12>
        <channel_13>0</channel_13>
        <channel_14>1</channel_14>
        <channel_15>3</channel_15>
        <channel_16>1</channel_16>
        <channel_17>0</channel_17>
        <channel_18>0</channel_18>
        <channel_19>0</channel_19>
        <channel_20>0</channel_20>
        <channel_21>0</channel_21>
        <channel_22>0</channel_22>
        <channel_23>0</channel_23>
        <channel_24>0</channel_24>
    </data_aggregator_ryg_point>
    ...
</data_aggregator_ryg_data>
```


CSV

```
Timestamp,Channel 1,Channel 2,Channel 3,Channel 4,Channel 5,Channel 6,Channel 7,Channel 8,Channel 9
1499644812000,6,3,1,1,2,1,1,1,3,1,1,1,0,1,3,1,0,0,0,0,0,0,0,0
1499644813000,6,3,1,1,2,1,1,1,3,1,1,1,0,1,0,1,0,0,0,0,0,0,0,0
...
```

B.3.3 Cabinet RYG Phase Data

B.3.3.1 About

This outputs all of the phase data for the time frame requested for the sensor requested if phase information has been previously configured on this sensor.

B.3.3.2 Endpoint

`/datastream/da/ryg/phase/[FORMAT]/[API_KEY]/[DEVICE_ID]/[START_TIME]/[END_TIME]`

B.3.3.3 Parameters

- **[FORMAT]**
  The format of the data you are querying, either ”json”, ”xml”, or ”csv”
- **[API_KEY]**
  - A generated user access token used to authenticate a user against the Acyclica Go software. Contact support@acyclica.com to receive your API KEY.
- **[DEVICE_ID]**
  - The sensor ID as noted in the enumeration query.
- **[START_TIME]**
  - The beginning time in the range you are searching. Time values must be in epoch format, in **milliseconds**. March 15, 2017 00:00 would be represented by `1489536000000`.
- **[END_TIME]**
  - The end time in the range you are searching. Time values must be in epoch format, in **milliseconds**. March 15, 2017 03:00 would be represented by `1489546800000`.

B.3.3.4 Example Output

JSON

```
[
    {
        "ryg_data": [
            {
                "ryg_id": 1,
                "phase": 2,
                "channel": 1
            },
            {
                "ryg_id": 3,
                "phase": 3,
                "channel": 2
            },
            {
                "ryg_id": 3,
                "channel": 3
            },
            {
                "ryg_id": 1,
                "phase": 5,
                "channel": 4
            },
            {
                "ryg_id": 1,
                "phase": 6,
                "channel": 5
            },
            {
                "ryg_id": 3,
                "phase": 7,
                "channel": 6
            },
            {
                "ryg_id": 3,
                "phase": 9,
                "channel": 7
            },
            {
                "ryg_id": 1,
                "phase": 4,
                "channel": 8
            },
            {
                "ryg_id": 3,
```

```
                "channel": 9
            },
            {
                "ryg_id": 1,
                "phase": 8,
                "channel": 10
            },
            {
                "ryg_id": 3,
                "channel": 11
            },
            {
                "ryg_id": 1,
                "channel": 12
            },
            {
                "ryg_id": 1,
                "channel": 13
            },
            {
                "ryg_id": 1,
                "channel": 14
            },
            {
                "ryg_id": 1,
                "channel": 15
            },
            {
                "ryg_id": 1,
                "channel": 16
            },
            {
                "ryg_id": 0,
                "channel": 17
            },
            {
                "ryg_id": 0,
                "channel": 18
            },
            {
                "ryg_id": 0,
                "channel": 19
            },
            {
                "ryg_id": 0,
```

```
                "channel": 20
            },
            {
                "ryg_id": 0,
                "channel": 21
            },
            {
                "ryg_id": 0,
                "channel": 22
            },
            {
                "ryg_id": 0,
                "channel": 23
            },
            {
                "ryg_id": 0,
                "channel": 24
            }
        ],
        "timestamp": 1483552800000
    },
    ...
]
```


XML


```
<data_aggregator_ryg_data>
    <data_aggregator_ryg_point>
        <timestamp>1483552800000</timestamp>
        <channel_1>
            <ryg_id>1</ryg_id>
            <phase>2</phase>
        </channel_1>
        <channel_2>
            <ryg_id>3</ryg_id>
            <phase>3</phase>
        </channel_2>
        <channel_3>
           <ryg_id>3</ryg_id>
        </channel_3>
        <channel_4>
            <ryg_id>1</ryg_id>
            <phase>5</phase>
        </channel_4>
        <channel_5>
            <ryg_id>1</ryg_id>
```

```
        <phase>6</phase>
    </channel_5>
    <channel_6>
        <ryg_id>3</ryg_id>
        <phase>7</phase>
    </channel_6>
    <channel_7>
        <ryg_id>3</ryg_id>
        <phase>9</phase>
    </channel_7>
    <channel_8>
        <ryg_id>1</ryg_id>
        <phase>4</phase>
    </channel_8>
    <channel_9>
        <ryg_id>3</ryg_id>
    </channel_9>
    <channel_10>
        <ryg_id>1</ryg_id>
        <phase>8</phase>
    </channel_10>
    <channel_11>
        <ryg_id>3</ryg_id>
    </channel_11>
    <channel_12>
        <ryg_id>1</ryg_id>
    </channel_12>
    <channel_13>
        <ryg_id>1</ryg_id>
    </channel_13>
    <channel_14>
        <ryg_id>1</ryg_id>
    </channel_14>
    <channel_15>
        <ryg_id>1</ryg_id>
    </channel_15>
    <channel_16>
        <ryg_id>1</ryg_id>
    </channel_16>
    <channel_17>
        <ryg_id>0</ryg_id>
    </channel_17>
    <channel_18>
        <ryg_id>0</ryg_id>
    </channel_18>
    <channel_19>
```

```
            <ryg_id>0</ryg_id>
        </channel_19>
        <channel_20>
            <ryg_id>0</ryg_id>
        </channel_20>
        <channel_21>
            <ryg_id>0</ryg_id>
        </channel_21>
        <channel_22>
            <ryg_id>0</ryg_id>
        </channel_22>
        <channel_23>
            <ryg_id>0</ryg_id>
        </channel_23>
        <channel_24>
            <ryg_id>0</ryg_id>
        </channel_24>
    </data_aggregator_ryg_point>
    ...
</data_aggregator_ryg_data>
```


B.4 Data Aggregator 300

B.4.1 Data Aggregator Sensor Listing

B.4.1.1 About

This enumerates the DA-300 units you have access to view, along with any mapped RYG channels, if they are configured.

B.4.1.2 Endpoint

`/datastream/da/meta/[FORMAT]/[API_KEY]`

B.4.1.3 Parameters

- **[FORMAT]**
  - The format of the data you are querying, either "xml" or "json"
- **[API_KEY]**
  - A generated user access token used to authenticate a user against the Acyclica Go software. Contact support@acyclica.com to receive your API KEY.

B.4.1.4 Example Output

XML

```
<data_aggregators>
    <data_aggregator>
        <id>[ID]</id>
        <reference_id>[DA_Serial]</reference_id>
        <cabinet_id>[ID]</cabinet_id>
        <ryg_map></ryg_map>
    </data_aggregator>
    <data_aggregator>
        <id>[ID]</id>
        <reference_id>[DA_Serial]</reference_id>
        <cabinet_id>[ID]</cabinet_id>
        <ryg_map>
            <ryg_channel>
                <channel>2</channel>
                <phase>3</phase>
            </ryg_channel>
            <ryg_channel>
                <channel>3</channel>
                <phase>3</phase>
            </ryg_channel>
            <ryg_channel>
                <channel>6</channel>
                <phase>5</phase>
            </ryg_channel>
            <ryg_channel>
                <channel>7</channel>
                <phase>5</phase>
            </ryg_channel>
            <ryg_channel>
                <channel>5</channel>
                <phase>4</phase>
            </ryg_channel>
            <ryg_channel>
                <channel>8</channel>
                <phase>4</phase>
            </ryg_channel>
        </ryg_map>
    </data_aggregator>
    ...
</data_aggregators>
```

JSON

```
[
    {
        "cabinet_id": 1,
        "data_aggregator_reference_id": "[DA_Serial]",
        "data_aggregator_id": 1,
        "ryg_map": []
    },
    {
        "cabinet_id": 2,
        "data_aggregator_reference_id": "[DA_Serial]",
        "data_aggregator_id": 2,
        "ryg_map": [
            {
                "channel": 2,
                "phase": 3
            },
            {
                "channel": 3,
                "phase": 3
            },
            {
                "channel": 6,
                "phase": 5
            },
            {
                "channel": 7,
                "phase": 5
            },
            {
                "channel": 5,
                "phase": 4
            },
            {
                "channel": 8,
                "phase": 4
            }
        ]
    },
    {
        ...
    }
]
```

B.5 OD Groups

B.5.1 Origin-Destination Group Listing

B.5.1.1 About

This will enumerate all of the sensors in each of the Origin/Destination Groups your user may access.

B.5.1.2 Endpoint

`/datastream/odgroup/inventory/xml/[API_KEY]`

B.5.1.3 Parameters

- **[API_KEY]**
  - A generated user access token used to authenticate a user against the Acyclica Go software. Contact support@acyclica.com to receive your API KEY.

B.5.1.4 Example Output

XML


```xml
<AcyclicaODInventory>
    <AcyclicaODGroup>
        <Name>Origin Destination Group Name</Name>
        <ODGroupID>OD Group ID</ODGroupID>
        <AcyclicaSensors>
            <AcyclicaSensor>
                <Serial>Sensor 1 Serial</Serial>
                <Name>Sensor 1 Name</Name>
                <Latitude>Sensor 1 Lat</Latitude>
                <Longitude>Sensor 1 Lon</Longitude>
            </AcyclicaSensor>
            <AcyclicaSensor>
                <Serial>Sensor 2 Serial</Serial>
                <Name>Sensor 2 Name</Name>
                <Latitude>Sensor 2 Lat</Latitude>
                <Longitude>Sensor 2 Lon</Longitude>
            </AcyclicaSensor>
            ...
        </AcyclicaSensors>
        ...
</AcyclicaODGroup>
```

B.6 Route

B.6.1 Route Listing

B.6.1.1 About

This will enumerates all routes available to your user in an XML format.

B.6.1.2 Endpoint

`/datastream/route/inventory/xml/[API_KEY]`

B.6.1.3 Parameters

- **[API_KEY]**
  - A generated user access token used to authenticate a user against the Acyclica Go software. Contact support@acyclica.com to receive your API KEY.

B.6.1.4 Example Output

XML

```
<AcyclicaRouteInventory>
    <AcyclicaRoute>
        <Name>Route 1 Name</Name>
        <RouteID>1</RouteID>
        <AcyclicaSegments>
            <AcyclicaSegment position="0">
                <Start>Sensor ID 1</Start>
                <End>Sensor ID 2</End>
            </AcyclicaSegment>
        </AcyclicaSegments>
    </AcyclicaRoute>
    <AcyclicaRoute>
        <Name>Route 2 Name/Name>
        <RouteID>2</RouteID>
        <AcyclicaSegments>
            <AcyclicaSegment position="0">
                <Start>Sensor ID 3</Start>
                <End>Sensor ID 4</End>
            </AcyclicaSegment>
            ...
            <AcyclicaSegment position="4">
                <Start>Sensor ID 11</Start>
                <End>Sensor ID 12</End>
            <AcyclicaSegment>
```

```
        </AcyclicaSegments>
    </AcyclicaRoute>
    ...
</AcyclicaRouteInventory>
```

B.6.2 Route Data

B.6.2.1 About

This will provide you with basic information about a route for a given time frame. The algorithms used are organized as follows:

1. Strength
2. First
3. Last
4. Minimum
5. Last

B.6.2.2 Endpoint

`/datastream/route/[FORMAT]/time/[API_KEY]/[ROUTE_ID]/[START_TIME]/[END_TIME]`

B.6.2.3 Parameters

- **[FORMAT]**
  The format of the data you are querying, either "json", "xml", or "csv"
- **[API_KEY]**
  - A generated user access token used to authenticate a user against the Acyclica Go software. Contact support@acyclica.com to receive your API KEY.
- **[ROUTE_ID]**
  - The route identifier, found in Acyclica Go or in the Route Listing API call.
- **[START_TIME]**
  - The beginning time in the range you are searching. Time values must be in epoch format, in **seconds**. March 15, 2017 00:00 would be represented by `148953600`.
- **[END_TIME]**
  - The end time in the range you are searching. Time values must be in epoch format, in **seconds**. March 15, 2017 03:00 would be represented by `148954680`.

B.6.2.4 Example Output

XML

```
<TrafficRoute ID="Route ID" Owner="">
    <HardwareType/>
    <FirmwareType>Independent</FirmwareType>
    <FirmwareVersion/>
    <Group/>
        <Route ID="Route ID">
        <HardwareType>Acyclica Probe</HardwareType>
        <Name>Route Name</Name>
        <Description>Route Description</Description>
        <Data>
            <Time format="unix" zone="utc" capture="first">1489104001816</Time>
            <Record Algorithm="1" Elapsed_time="632728"/>
            <Record Algorithm="2" Elapsed_time="604255"/>
            <Record Algorithm="3" Elapsed_time="641072"/>
            <Record Algorithm="4" Elapsed_time="537571"/>
            <Record Algorithm="5" Elapsed_time="677242"/>
        </Data>
        <Data>
            <Time format="unix" zone="utc" capture="first">1489104004478</Time>
            <Record Algorithm="1" Elapsed_time="632728"/>
            <Record Algorithm="2" Elapsed_time="604255"/>
            <Record Algorithm="3" Elapsed_time="638474"/>
            <Record Algorithm="4" Elapsed_time="536917"/>
            <Record Algorithm="5" Elapsed_time="677242"/>
        </Data>
        ...
    </Route>
</TrafficRoute>
```

CSV

```
Timestamp,Strengths,Firsts,Lasts,Minimums,Maximums
1489104001816,632728,604255,641072,537571,677242
1489104004478,632728,604255,638474,536917,677242
1489104008260,632728,604255,638474,536917,677242
...
```

B.6.3 Route Information -- Most Recent Data

B.6.3.1 About

This will return the last reported data for a given route in an XML format.

B.6.3.2 Endpoint

`/datastream/route/xml/time/last/[API_KEY]/[ROUTE_ID]`

B.6.3.3 Parameters

- **[API_KEY]**
  - A generated user access token used to authenticate a user against the Acyclica Go software. Contact support@acyclica.com to receive your API KEY.
- **[ROUTE_ID]**
  - The route identifier, found in Acyclica Go or in the Route Listing API call.

B.6.3.4 Example Output

XML


```xml
<TrafficRoute ID="Route ID" Owner="">
    <HardwareType/>
    <FirmwareType>Independent</FirmwareType>
    <FirmwareVersion/>
    <Group/>
    <Route ID="Route ID">
        <HardwareType>Acyclica Probe</HardwareType>
        <Name>Route Name</Name>
        <Description/>
        <Data>
            <Time format="unix" zone="utc" capture="first">1489557570797</Time>
            <Record Algorithm="1" Elapsed_time="369112"/>
            <Record Algorithm="2" Elapsed_time="343179"/>
            <Record Algorithm="3" Elapsed_time="369395"/>
            <Record Algorithm="4" Elapsed_time="321634"/>
            <Record Algorithm="5" Elapsed_time="408376"/>
        </Data>
    </Route>
</TrafficRoute>
```

B.7 VSO Sensors

B.7.1 VSO Listing

B.7.1.1 About

This enumerates the VSO sensors available to your user account.

B.7.1.2 Endpoint

`/datastream/vso/meta/[FORMAT]/[API_KEY]`

B.7.1.3 Parameters

- **[FORMAT]**
  - The format of the data you are querying, either "json" or "xml"
- **[API_KEY]**
  - A generated user access token used to authenticate a user against the Acyclica Go software. Contact support@acyclica.com to receive your API KEY.

B.7.1.4 Example Output

JSON


```
[
    {
        "description": "VSO Sensor Description",
        "name": "VSO Sensor Name",
        "zones": [
            {
                "zone_id": ZONEID,
                "reference_id": "REFERENCE ID"
            },
            {
                ...
            }
        ],
        "id": VSOID,
        "lastdata": 1489611839000,
        "count_stations": [
            {
                "route_id": 4314,
                "name": "alpha",
                "zones": [
```

```
                    1,
                    2,
                    3
                ],
                "cs_id": 3
            }
        ],
        "location_id": LOCATIONID,
        "client_groups": [
            6
        ],
        "dev_type": "Sensor Type",
        "short_name": "",
        "serial": "SERIAL"
    },
    {
        ...
    }
]
```


XML


```
<vso_sensors>
    <vso_sensor>
        <id>1</id>
        <serial>306</serial>
        <name>Office 2</name>
        <short_name/>
        <description/>
        <dev_type>Sensys</dev_type>
        <latitude/>
        <longitude/>
        <lastdata/>
        <zones></zones>
        <count_stations></count_stations>
    </vso_sensor>
    ...
</vso_sensors>
```

B.7.2 VSO Count Station Data

B.7.2.1 About

This returns the data for a VSO sensor count station over the given time frame, using the given period, in the requested format.

B.7.2.2 Endpoint

```
/datastream/vso/count_station_data/[FORMAT]/time/[API_KEY]/
[DEVICE_ID]/[PERIOD]/[START_TIME]/[END_TIME]
```

B.7.2.3 Parameters

- **[FORMAT]**
  The format of the data you are querying, either "json", "xml", or "csv"
- **[API_KEY]**
  - A generated user access token used to authenticate a user against the Acyclica Go software. Contact support@acyclica.com to receive your API KEY.
- **[DEVICE_ID]**
  - The sensor ID as noted in the ennumeration query.
- **[PERIOD]**
  - This is an integer representing the millisecond period for which the data should be grouped. Must be larger than 60000 (ms).
- **[START_TIME]**
  - The beginning time in the range you are searching. Time values must be in epoch format, in **milliseconds**. March 15, 2017 00:00 would be represented by `1489536000000`.
- **[END_TIME]**
  - The end time in the range you are searching. Time values must be in epoch format, in **milliseconds**. March 15, 2017 03:00 would be represented by `1489546800000`.

B.7.2.4 Example Output

JSON


```
[
    {
        "speed": 0,
        "timestamp": 1483552700000,
        "occupancy": 0,
```

```
        "count_station_id": [VSO_ID],
        "volume": 0
    },
    {
        "speed": 0.0,
        "timestamp": 1483552700000,
        "occupancy": 0.0,
        "count_station_id": [VSO_ID],
        "volume": 16
    },
    ...
]
```


XML


```
<vso_sensor>
    <vso_count_station_data time="1483552700000">
        <count_station_id>[VSO_ID]</count_station_id>
        <volume>0</volume>
        <speed>0</speed>
        <occupancy>0</occupancy>
    </vso_count_station_data>
    <vso_count_station_data time="1483552700000">
        <count_station_id>[VSO_ID]</count_station_id>
        <volume>16</volume>
        <speed>0.0</speed>
        <occupancy>0.0</occupancy>
    </vso_count_station_data>
    ...
</vso_sensor>
```


B.7.3 VSO Zone Data

B.7.3.1 About

This returns data formatted as requested for the zone data for the VSO sensor requested over the time frame requested, using the period requested.

B.7.3.2 Endpoint

`/datastream/vso/zone_data/json/time/[API_KEY]/[DEVICE_ID]/[PERIOD]/[START_TIME]/[END_TIME]`

B.7.3.3 Parameters

- **[FORMAT]**
  The format of the data you are querying, either ”json”, ”xml”, or ”csv”
- **[API_KEY]**
  - A generated user access token used to authenticate a user against the Acyclica Go software. Contact support@acyclica.com to receive your API KEY.
- **[DEVICE_ID]**
  - The sensor ID as noted in the ennumeration query.
- **[PERIOD]**
  - This is an integer representing the millisecond period into which the data should be grouped. Must be larger than 60000 (ms).
- **[START_TIME]**
  - The beginning time in the range you are searching. Time values must be in epoch format, in **seconds**. March 15, 2017 00:00 would be represented by `1489536000`.
- **[END_TIME]**
  - The end time in the range you are searching. Time values must be in epoch format, in **seconds**. March 15, 2017 03:00 would be represented by `1489546800`.

B.7.3.4 Example Output

JSON


```
[
    {
        "speed": 0,
        "timestamp": 1489536000000,
        "zone_id": [ZONE_ID],
        "volume": 0,
        "occupancy": 0
    },
    {
        "speed": 0,
        "timestamp": 1489536000000,
        "zone_id": [ZONE_ID],
        "volume": 0,
        "occupancy": 0
    },
    ...
]
```

B.8 Segments

B.8.1 Segment Match Data

B.8.1.1 About

This returns the total time for a match to progress through a segment of two adjacent sensors in a segment on a route for all available algorithms. The algorithms are as follows:

1. Strength
2. First
3. Last
4. Minimum
5. Maximum

B.8.1.2 Endpoint

`/datastream/segment/[FORMAT]/time/[API_KEY]/[DEVICE_ID1]/[DEVICE_ID2]/[START_TIME]/[END_TIME]`

B.8.1.3 Parameters

- **[FORMAT]**
  The format of the data you are querying, either ”xml” or ”csv”
- **[API_KEY]**
  - A generated user access token used to authenticate a user against the Acyclica Go software. Contact support@acyclica.com to receive your API KEY.
- **[DEVICE_ID1]**
  - The serial for the first sensor in a segment, must be adjacent to sensor 2.
- **[DEVICE_ID2]**
  - The serial for the first sensor in a segment, must be adjacent to sensor 1.
- **[START_TIME]**
  - The beginning time in the range you are searching. Time values must be in epoch format, in **seconds**. March 15, 2017 00:00 would be represented by `1489536000`.
- **[END_TIME]**
  - The end time in the range you are searching. Time values must be in epoch format, in **seconds**. March 15, 2017 03:00 would be represented by `1489546800`.

B.8.1.4 Example Output

XML

```xml
<TrafficSegment ID="[SEGMENT_ID]" Owner="">
    <HardwareType/>
    <FirmwareType>Independent</FirmwareType>
    <FirmwareVersion/>
    <Group/>
    &lt;Segment ID="[SEGMENT_ID]"&gt;
        <HardwareType>Acyclica Probe</HardwareType>
        <Name/>
        <Description/>
        <Data>
            <Time format="unix" zone="utc" capture="first">1489104019.299</Time>
            <Hash>
                [MATCH_1]
            </Hash>
            <Elapsed_time>43.647</Elapsed_time>
            <Algorithm>1</Algorithm>
        </Data>
        <Data>
            <Time format="unix" zone="utc" capture="first">1489104019.299</Time>
            <Hash>
                [MATCH_2]
            </Hash>
            <Elapsed_time>10.0</Elapsed_time>
            <Algorithm>2</Algorithm>
        </Data>
        ...
    &lt;/Segment&gt;
</TrafficSegment>
```

B.8.2 Segment Match Algorithm Data

B.8.2.1 About

This gives the five algorithms for matches between a given time range for the match to progress through a segment of two adjacent sensors in a segment of a route.

B.8.2.2 Endpoint

/datastream/segment/v2/xml/time/[API_KEY]/[DEVICE_ID1]/[DEVICE_ID2]/[START_TIME]/[END_TIME]

B.8.2.3 Parameters

- **[API_KEY]**
  - A generated user access token used to authenticate a user against the Acyclica Go software. Contact support@acyclica.com to receive your API KEY.
- **[DEVICE_ID1]**
  - The serial for the first sensor in a segment, must be adjacent to sensor 2.
- **[DEVICE_ID2]**
  - The serial for the first sensor in a segment, must be adjacent to sensor 1.
- **[START_TIME]**
  - The beginning time in the range you are searching. Time values must be in epoch format, in **seconds**. March 15, 2017 00:00 would be represented by 1489536000.
- **[END_TIME]**
  - The end time in the range you are searching. Time values must be in epoch format, in **seconds**. March 15, 2017 03:00 would be represented by 1489546800.

B.8.2.4 Example Output

XML


```xml
&lt;segment id="[SEGMENT_ID]"&gt;
    <matchdata>
        <data>
            <time>1489104019.299</time>
            <strength>43.647</strength>
            <first>10.0</first>
            <last>114.995</last>
            <minimum>10.0</minimum>
            <maximum>114.995</maximum>
        </data>
        <data>
            <time>1489104000.573</time>
            <strength>489.062</strength>
            <first>549.969</first>
            <last>489.126</last>
            <minimum>488.978</minimum>
            <maximum>550.117</maximum>
        </data>
        ...
```

```
    &lt;/matchdata&gt;
&lt;/segment&gt;
```

B.8.3 Segment Match Filtered Algorithm Data

B.8.3.1 About

This returns a filtered CSV-formatted listing of the five algorithms for matches found having progressed between two adjacent sensors during the given time period.

B.8.3.2 Endpoint

`/datastream/segment/filtered/csv/time/[API_KEY]/[DEVICE_ID1]/[DEVICE_ID2]/[START_TIME]/[END_TIME]`

B.8.3.3 Parameters

- **[API_KEY]**
  - A generated user access token used to authenticate a user against the Acyclica Go software. Contact support@acyclica.com to receive your API KEY.
- **[DEVICE_ID1]**
  - The serial for the first sensor in a segment, must be adjacent to sensor 2.
- **[DEVICE_ID2]**
  - The serial for the first sensor in a segment, must be adjacent to sensor 1.
- **[START_TIME]**
  - The beginning time in the range you are searching. Time values must be in epoch format, in **seconds**. March 15, 2017 00:00 would be represented by `1489536000`.
- **[END_TIME]**
  - The end time in the range you are searching. Time values must be in epoch format, in **seconds**. March 15, 2017 03:00 would be represented by `1489546800`.

B.8.3.4 Example Output

CSV

```
Timestamp,MAC Hash,Elapsed,AlgoId
1489104017.628,2ac18e62c91b35003af5eda83fef94526d842df478e8bc8c51857686f0a5a020,150.457,1
1489104017.628,2ac18e62c91b35003af5eda83fef94526d842df478e8bc8c51857686f0a5a020,10.0,2
1489104017.628,2ac18e62c91b35003af5eda83fef94526d842df478e8bc8c51857686f0a5a020,10.0,3
```

```
1489104017.628,2ac18e62c91b35003af5eda83fef94526d842df478e8bc8c51857686f0a5a020,10.0,4
1489104017.628,2ac18e62c91b35003af5eda83fef94526d842df478e8bc8c51857686f0a5a020,150.457,5
...
```