Multimodal Data Set Clean-up for
Portland Oregon Metropolitan Region

Data Set Description and Dictionary

Arterial Data

April 12, 2012

Portland State UNIVERSITY

METRO

Oregon Department of Transportation

TRIMET See where it takes you.

CITY OF PORTLAND OREGON 1851

Table of Contents

1 Data Description

This document provides the data description for the PORTAL Arterial Data for the FHWA Test Data Set project submission. This document contains a general description of System Detector, Bluetooth and Phase and Timing data and meta-data tables. Figure 1 shows the schematic of the arterial data sources. In addition, we document naming conventions in those tables. Data collection period is Sept 15, 2011 through Nov 15, 2011.

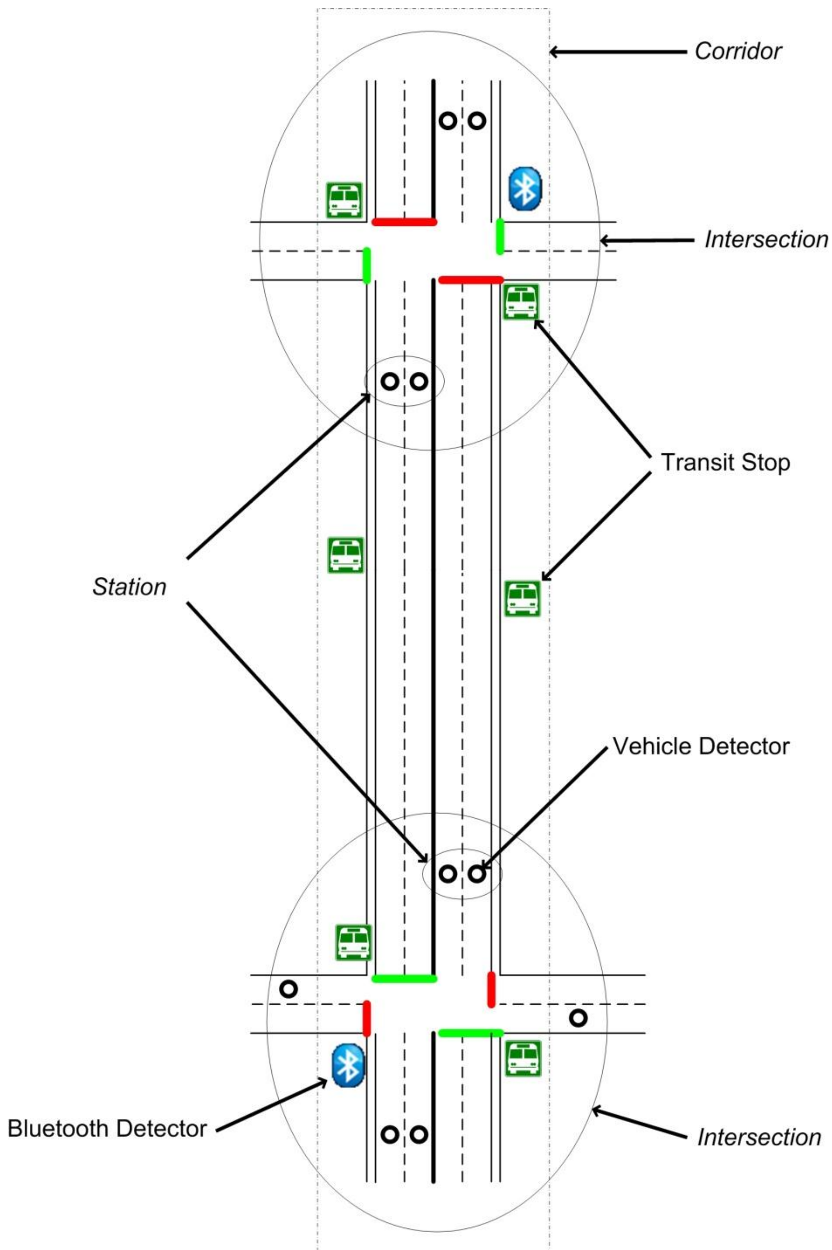

Figure 1 Arterial Data Sources

1.1 System Detectors

System detector data consists of volume and occupancy data collected approximately every minute from single loop detectors installed 300-400 ft upstream or downstream of a signalized intersection as seen in Figure 1. There are four sets of system detectors present along 82nd Avenue as shown in Figure 2.

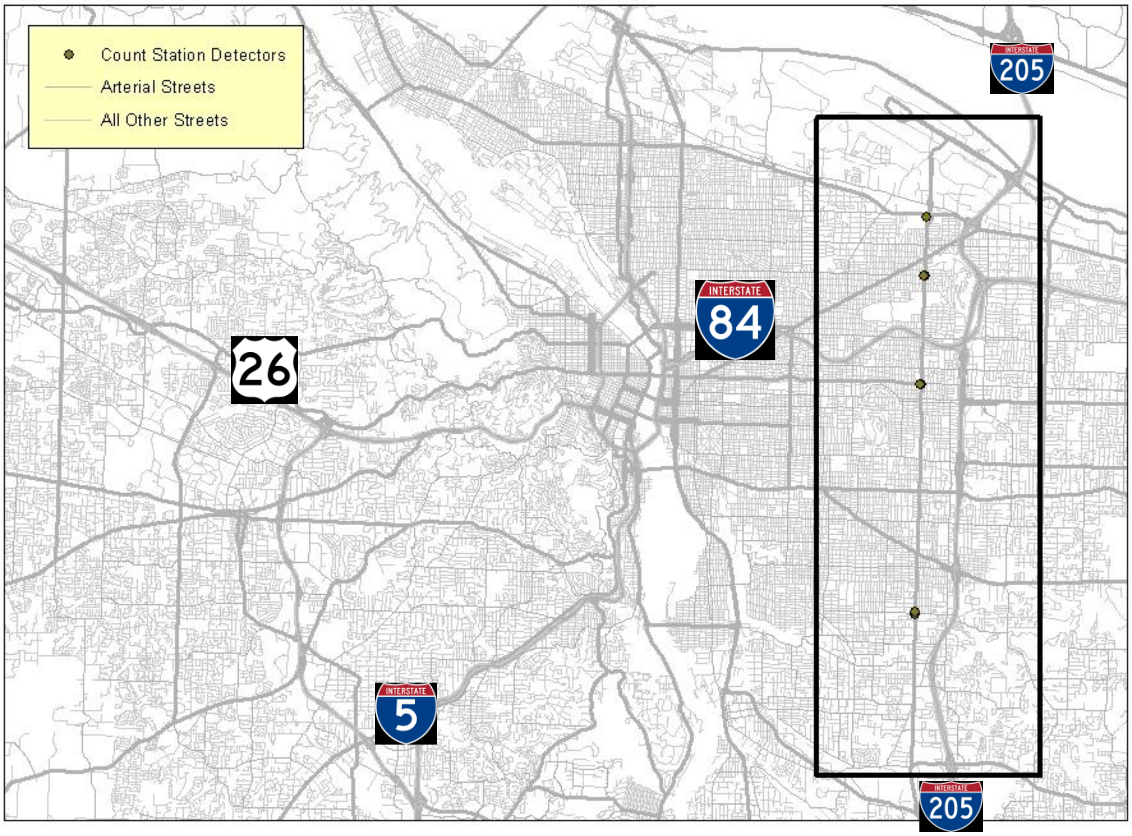

Figure 2 Arterial System Loop Detector Locations

1.1.1 Detectors, Stations and Intersections

A detector refers to an individual detector. A station is a set of related detectors. Stations are associated with intersections. Detectors are contained in stations and stations are contained in intersections.

1.1.2 Meta-Data (detectors, stations, intersections)

The system loop detectors on arterials are classified in a hierarchical organization consisting of detectors, stations, intersections. A detector refers to an individual loop detector. A station is a grouping of loop detectors by approach. Stations are contained in intersections. An intersection is a grouping of all detectors within the bounds of its functional area. The intersection grouping includes stations, transit stops and Bluetooth detectors.

1.1.3 Data Quality

Data quality flags in *raw_detector_archive* file are assigned for volume and occupancy values based on maximum and minimum theoretical limits. These tests are listed in Table 1. The tests apply to data aggregated by the sample period. The flags are Boolean and can either be true or false. Each test includes a condition; data samples that satisfy that condition are considered improbable and may indicate a malfunction and the flag is set to “true”. For example, a report of a 1 minute count greater than 30, which corresponds to a flow of over 1800 vehicles/hour/lane, is considered improbable.

The data quality flags appear as a Boolean field for each data record. The flags are either true or false. The data quality flags for volume are set to true if the volume is negative or greater than the theoretical maximum for the sample period (calculated based on saturation flow). Similarly the data quality flags for occupancy are set to true if the occupancy is negative or greater than 1000 (occupancy is recorded in tenths of a percent). Finally the visual inspection flag is set to true if the volumes or occupancies look improbable (for example, consistent high volumes (but lower than the max volume) at night).

Table 1 Detector Configuration Tests: Conditions

Condition	Condition Name	Flag
Volume < 0	DQ_MINVOL	T
Volume > (s*sat flow)/3600	DQ_MAXVOL	T
Occupancy < 0	DQ_MINOCC	T
Occupancy > 1000	DQ_MAXOCC	T
Visual Inspection	DQ_VISUAL	T

1.1.4 Outages

The following long outages occurred during the two month data collection period (Sept 15th – Nov 15th).

Table 2 Outages in the Raw Detector Archive File

Detectorid	Location	Start time	End time	Reason
409	82nd Ave and Burnside St	11/11/2011 0:00	11/15/2011 0:00	No data
410	82nd Ave and Burnside St	11/11/2011 0:00	11/15/2011 0:00	No data
411	82nd Ave and Burnside St	11/11/2011 0:00	11/15/2011 0:00	No data
412	82nd Ave and Burnside St	11/11/2011 0:00	11/15/2011 0:00	No data

1.2 Phase and Timing

Phase and timing data consists of active calls and phasing information for thirty two signals along the 82nd Avenue corridor that are operated by the City of Portland. Figure 3 shows the location of the signalized intersections along the 82nd Avenue corridor.

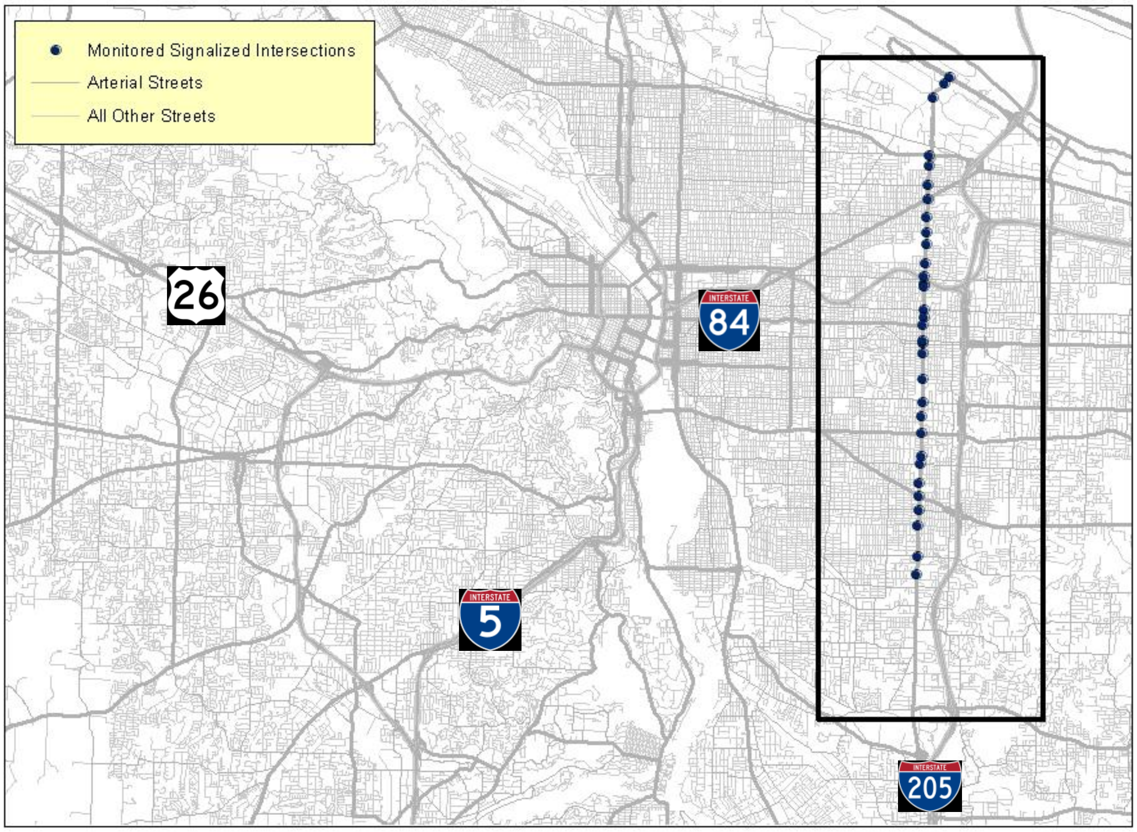

Figure 3 Signalized Intersection Locations

The fields in the signal phase and timing data file include timestamp, the id of the signal controller and the current timing plan number in effect. A series of bit flag fields are used to indicate phase information. Bit flag fields are a data compression technique that use the bits in an integer value to represent a series of yes/no flags. In computer hardware, an integer is often represented with 16 bits – each bit having a value of either 0 or 1. By setting each of the sixteen bits to 0 or 1 as appropriate, those 16 bits can be used to represent 16 yes/no flags. Similarly integer bit flags are used to represent phases with active vehicle and pedestrian calls as well as phases currently displaying yellow or green. For the Yellow field, the value 136 (128+8) indicates that phases 8 and 4 are currently displaying yellow. The final field represents the value of the local timer. Also included are online and status information and a bit flag ped field representing phases currently displaying a Pedestrian Walk indication.

Table 3 shows the values associated with each phase. To decode a value located in either greens, yellow, peds, ped_calls and veh_calls, you do the following:

1. Start with the value being decoded.
2. Choose a number from table 3 in the value column that is the largest number that is less than or equal to the value being decoded.
3. The corresponding phase number to that value is a phase that is currently active.
4. Subtract the value from the table from the value being decoded.
5. If there is a remainder, choose a value from the table that largest number that is less than or equal to the remainder value being decoded.
6. The corresponding phase number to the value from the table is active.
7. Repeat steps 4 through 6 until you are left with zero.

An example of the above steps are as follows:

1. A value of 196 is present in the “Greens” column.
2. The largest value in Table 3 that is less than or equal to 196 is 128, therefore phase 8 is currently active.
3. Subtracting 128 from 196 leaves 68.
4. The largest value in Table 3 that is less than or equal to 68 is 64, therefore phase 7 is currently active.
5. Subtracting 64 from 68 leaves 4.
6. The largest value in table 3 that is less than or equal to 4 is 4, therefore phase 3 is currently active.
7. Subtracting 4 from 4 leaves 0, therefore phases 8, 7, and 3 are currently active.

Table 3 Phases and Values

Phase Number	Value
1	1
2	2
3	4
4	8
5	16
6	32
7	64
8	128

Alternatively, the phases can also be read from Table 4 which shows the possible combinations of numbers in the greens, yellow, peds, ped_calls and veh_calls columns. In order to determine the phases that correspond to a certain value in the table, locate the value and the column heading associated with that value indicates the phase. Note that the same value can appear in multiple columns, which indicates that those corresponding phases are active together. Following the above example, a value of 196 indicates that phases 3, 7 and 8 are active.

Table 4 Possible Combinations of Values

Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Phase 6	Phase 7	Phase 8
1	2	4	8	16	32	64	128
3	3	12	12	17	33	68	132
17	18	68	72	18	34	72	136
33	34	132	136	48	48	192	192
19	19	76	76	19	35	76	140
35	35	140	140	49	49	196	196
49	50	196	200	50	50	200	200
51	51	204	204	51	51	204	204

An overlap is a controller output that is associated with two or more phases and is used to allow conflicting phases to be active at the same time (1). In the signal controller software used by the City of Portland, four overlaps are allowed. The Overlays field represents overlaps that are currently displaying a Green indication. Table 5 shows the values associated with the overlaps. For example a value of 1 in overlays indicates that the first overlap is displaying green; a value of 4 indicates the third overlap is displaying green.; a value of 10 indicates that the second and fourth overlaps are displaying green (10 = 8+2). Once the overlaps have been determined using Table 5, the phases associated with each overlap can be determined by looking at the overlaps table in appropriate signal timing plan for the associated intersection. Upon observation of the data, overlays are present only for two intersections (2115, 2146). Upon further examination it appears that only the overlays for intersection 2146 are currently functional. The phases associated with this overlap can be deduced using the timing file. Overlays (overlaps) for intersection 2115 are not functional and can be ignored.

Table 5 Overlays

Value	Overlaps
1	1
2	2
4	3
8	4

Timing plans for all intersections on the 82nd Avenue corridor are provided in the final data set. Each intersection has a timing plan with the following naming convention - intersectionID and Intersection Location. pdf. For example, the timing plan for intersection 2114 is named *as 2114 82nd & Wasco. pdf*. The intersectionId's correspond to the ID's in the arterial_intersections.csv file. The signal timing plans contain numerous tables pertaining to various parameters in the controller. The most relevant tables from signal timing perspective and those that would be most relevant to researchers are described here. Phase times on page 2 indicate the timing parameters for each phase. The Flashing Yellow Left Turn Arrow table on page 8 indicates phases (if any) that have flashing left turn arrows. The service plans tables on pages 9-10 indicate the timing plans that are active at a particular intersection. The coordination plans table on pages 12-13 outlines the timing parameters corresponding to each plan such as cycle lengths, offsets, force offs and splits. The time of day table on page 15 indicates the schedule for time of day plans.

1.3 Bluetooth

The Bluetooth data consists of travel times derived from matching Media Access Control (MAC) addresses that are captured by the Bluetooth readers. indication. MAC addresses are unique 48 bit addresses that are assigned to electronic devices by the manufacturers of these devices. Matching the MAC addresses between a pair of locations yields travel time between the two locations. A timestamp and a segment id are also present for each travel time record. At the onset of this project, two Bluetooth readers were present along 82nd Avenue. Midway through the project two additional readers were installed by the City of Portland; data for the two additional readers is not included in the data set since they were functional for only part of the project time period. Figure 4 shows the location of the two Bluetooth readers for which data is included in the data set.

Figure 4 Bluetooth Reader Locations

1.3.1 Meta-Data (detectors, intersections)

The Bluetooth readers on arterials are classified in a hierarchical organization consisting of detectors, stations and intersections. A detector refers to an individual Bluetooth reader. A station is a grouping of Bluetooth readers at a particular location. Stations are contained in intersections. An intersection is a grouping of all detectors within the bounds of its functional area. This classification is similar to the classification of system detector meta-data described in Section 1.1.2.

1.3.2 Bluetooth Travel Times and Matching Algorithm

Collecting travel times using Bluetooth technology is gaining traction. Travel times are collected using Bluetooth technology by logging MAC addresses that are assigned to electronic devices in vehicles passing particular locations where readers are installed.

There are two data collection units (DCU) along 82nd Avenue at the intersections of 82nd and Woodward and 82nd and Foster. Each Bluetooth based DCU collects MAC address, date and time respectively (2). An algorithm has been developed to match unique MAC addresses between the two locations. Once an algorithm detects a unique MAC ID, it searches for the same MAC ID at the other location within a certain time range. The time range is set at twenty minutes so as to filter out trips with stops and to differentiate multiple trips occurring during a given time period. Based on the speed of the vehicle and the range of detection of the DCU, a single MAC address can be read multiple times at a single location as the vehicle passes by the reader. Therefore to get accurate travel times, matching can be performed in one of two ways: first – first match or last- last match. Our algorithm matches the MAC addresses based on last - last detection. Once a match has been found, the algorithm calculates travel time as the difference between the recorded timestamps at both locations.

The algorithm also filters the travel times for outliers. Each travel time value is checked against the previous ten readings and is flagged as an outlier if the current value exceeds the mean of the previous ten readings plus one standard deviation (3). Outliers are discarded from the final travel times file. The final Bluetooth travel times file contains a list of all matched travel times, from and to Bluetooth id's and a timestamp. Raw MAC addresses are not included in the final data file. The code for matching MAC addresses and filtering outliers is presented below. Pseudo-code for the matching algorithms is found in Appendix A.

1.3.3 Data Quality

During the time period between 9/15/11 and 10/17/11, there appears to have been a clock drift and clock mismatch at the two Bluetooth detector locations resulting in very low travel times between the two detector locations. As a result, the travel times between 9/15/11 and 10/17/11 for both directions (10001 -10002, 10002 – 10001) have been flagged as suspect. Also, there appears to be additional missing data from Foster – Woodward (10002 – 10001) during the same time. Suspect values are flagged as T in the dq_tt column in the Bluetooth_traveltimes.csv file.

Upon visual inspection (plots), we estimate that the addition of 100 seconds to the travel time values between Woodward – Foster (10001 – 10002) during the affected time range 9/15/11 0:00:00 – 10/17/11 12:00:00 would address the clock drift issue and would result in reasonable travel times for that direction that could be used for analysis. This additional 100 seconds is not added to the data in the data files; whether to discard the data during this time period or add the 100 seconds and use the data is a decision left to users of the data.

1.3.4 Outages

Table 6 shows the outages that occurred with the Bluetooth data during the two month collection period (9/15/11 – 11/15/11).

Table 6 Outages in the Bluetooth Travel Times File

From Bluetooth ID – To Bluetooth ID	Location	Start Time	End Time	Reason
10001 – 10002	Woodward - Foster	9/16/11 10:16	9/19/11 7:25	No Data
10001 – 10002	Woodward - Foster	10/3/11 9:40	10/4/11 12:45	No Data
10002 – 10001	Foster - Woodward	9/16/11 10:10	9/19/11 7:18	No Data
10002 – 10001	Foster - Woodward	10/3/11 9:39	10/4/11 12:23	No Data

2 Data File Listing and Description

Table 7 List of Arterial System Detector Data Files

File Name	Primary Key	Description
arterial_detectors.csv	detectorid	Meta-data for each detector in the data set. A detector is defined as a single dual-loop detector. Table provides the data dictionary for the detectors table.
arterial_intersections.csv	intersectionid	Meta-data for each intersection in the data set. Table 9 provides the data dictionary for the intersections table.
arterial_stations.csv	stationid	Meta-data for each station in the data set. A station is a set of detectors at a particular location, for example the detectors in all lanes at a particular location. Table provides the data dictionary for the stations table.
raw_detector_archive.csv	detectorid	Primary loop detector data table. Contains one minute volume, occupancy and data quality flags for the arterial loop detector data. Table 11 provides the data dictionary for the raw detector archive table.
phase_and_timing_data.csv	intersectionid	Primary signal timing and phase data table. Contains bit flags for phases that are currently displaying green, yellow, ped call and vehicle call indications.

		Table provides the data dictionary for the phase and timing data.
bluetooth_stations.csv	bluetoothid	Meta-data for the Bluetooth stations. A station is a set of detectors at a particular location. Table provides the data dictionary for the Bluetooth data.
bluetooth_traveltimes.csv	from_bluetooth_id, to_bluetooth_id, timestamp	Primary travel time table. Contains matched travel times between pairs of Bluetooth detectors. Table 1 provides the data dictionary for the Bluetooth travel time table.

3 Data Dictionaries

3.1 System Detectors

Table 8 Data Dictionary for Arterial Detector File (arterial_detectors.csv)

Attribute Name	Attribute Type	Description
detectorid	Integer	Id of the detector (key)
lane	Character (30)	Lane in which the detector is located. (CENTER, N/A, RIGHT, CENTER 1, BIKE, LEFT TURN, RAMP, CENTER 2, ON RAMP, LEFT)
stationid	Integer	Id of the station

Sample

Attribute Name	Example 1	Example 2
detectorid	253	254
lane	RIGHT	LEFT
stationid	156	156

Table 9 Data Dictionary for Arterial Intersection File (arterial_intersections.csv)

Attribute Name	Attribute Type	Description
intersectionid	Integer	Id of the intersection (Key)
location	Character (100)	Intersection name
lat	Character (30)	Latitude of the Intersection
lon	Character (30)	Longitude of the Intersection

Sample

Attribute Name	Example 1	Example 2
intersectionid	2117	4110
location	NE Fremont St & 82nd Ave	SE Powell Blvd & 82nd Ave
lat	45.548142	45.497411
lon	-122.578737	-122.578702

Table 10 Data Dictionary for Arterial Station File (arterial_stations.csv)

Attribute Name	Attribute Type	Description
stationid	Integer	ID of the station (Key)
bound	Character (1)	Direction of travel based on station location (N, S, E or W)
intersectionid	Integer	Id of the associated intersection
lat	Character (30)	Latitude of the Station
lon	Character (30)	Longitude of the Station

Sample

Attribute Name	Example1	Example 2
stationid	135	136
bound	N	S
intersectionid	6012	6012
lat	45.5226	45.522595
lon	-122.578983	-122.579149

Table 11 Data Dictionary for Raw Detector Archive File (raw_detector_archive.csv)

Attribute Name	Attribute Type	Description
detectorid	integer	ID of the detector (Key)
timestamp	timestampwithtimezone	Upload date and time
status	text	Displays the upload status (Good, Timeout, Bad Response)
sampleperiod	integer	Duration of time (seconds) during which data was collected
volume	integer	Volume in vehicles per sample period
occupancy	integer	Occupancy in units of tenth of a percent
dq_minvol	boolean	Data quality flag check for min volume; T if vol < 0
dq_maxvol	boolean	Data quality flag check for max volume; T if vol > (sampleperiod *sat flow)/3600
dq_minocc	boolean	Data quality flag check for min occupancy; T if occ < 0
dq_maxocc	boolean	Data quality flag check for max occupancy; T if occ > 1000
dq_visual	boolean	Data quality flag check for visual inspection of volume; T if volume data is suspect

Sample

Attribute Name	Example 1	Example 2
detectorid	253	255

timestamp	9/15/2011 1:20:59	9/20/2011 6:15:34
status	GOOD	GOOD
sampleperiod	15	15
sequencenumber	129	155
volume	1	0
occupancy	0	0
dq_minvol	f	F
dq_maxvol	f	F
dq_minocc	f	F
dq_maxocc	f	F
dq_visual	f	T

3.2 Phase and Timing

Table 12 Data Dictionary for Phase and Timing Data File (phase_and_timing_data.csv)

Attribute Name	Attribute Type	Description
fromtopofcycle	integer	the value of the local timer (i.e. the number of seconds elapsed after local zero)
greens	integer	Contains bit flags representing the Phases that are currently displaying a Green indication.
yellow	integer	Contains bit flags representing the Phases that are currently displaying a Yellow indication.
peds	integer	Contains bit flags representing the Phases that are currently displaying a Pedestrian Walk indication.
ped_calls	integer	Contains bit flags representing the Phases that have active ped calls.
veh_calls	integer	Contains bit flags representing the Phases that have active vehicle calls.
status	smallint	Indicates signal status. 0 – Normal, 1 - Preempt, 2- Transition, 3 – Flash, 4 – Free, 6 – Stop
online	smallint	whether or not the light controller is 'online'. 1 = Online, 0 = Offline.
intersectionid	integer	Id of the intersection
overlays	integer	Contains bit flags representing the overlaps that are currently displaying a Green indication. The least significant bit represents Overlap 1, the next least significant bit represents Overlap 2 and so on. If a bit is set, the corresponding Overlap is currently displaying Green.
plan_num	integer	the current timing plan number in effect
timestamp	timestamp	the time at which this data was recorded

Sample

Attribute Name	Example 1	Example 2
fromtopofcycle	1	62
greens	32	34
yellow	2	0
peds	32	34
ped_calls	42	34
veh_calls	3	132
status	2	2
online	1	1
intersectionid	4107	4109
overlays	0	0
plan_num	4	4
timestamp	9/15/2011 12:00:00 AM	9/15/2011 12:00:00 AM

3.3 Bluetooth

Table 13 Data Dictionary for Bluetooth Stations File (bluetooth_stations.csv)

Attribute Name	Attribute Type	Description
bluetoothid	integer	ID of the detector (Key)
intersectionid	integer	ID of the associated intersection
lat	Character (30)	Latitude of the bluetooth detector location
lon	Character (30)	Longitude of the bluetooth detector location

Sample

Attribute Name	Example 1	Example 2
bluetoothid	10001	10002
intersectionid	4117	4113
lat	45.5013	45.482784
lon	-122.5790	-122.5790

Table 14 Data Dictionary for Bluetooth Travel Times File (bluetooth_traveltimes.csv)

Attribute Name	Attribute Type	Description
from_bluetooth_id	integer	Id of the detector at the start of the segment (Key)
to_bluetooth_id	integer	Id of the detector at the end of the segment (Key)
timestamp	Timestamp with date	Date and time of the observed travel time (Key)
traveltime	integer	Matched travel time between a pair of Bluetooth detectors (units: seconds)
dq_tt	boolean	Data quality flag check for travel times; T if data is bad, F if data is good

Sample

Attribute Name	Example 1	Example 2
from_ bluetooth_id	10001	10002
to_bluetooth_id	10002	10001
timestamp	10/20/11 13:25	11/06/2011 20:19:03
traveltime	427	133
dq_tt	F	F

4 References

1. Federal Highway Administration. Traffic Signal Timing Manual. June 2008.

2. Kim, S., Porter, D., Magaña, M., Park, S. and A. Saeedi. Wireless Data Collection System for Travel Time Estimation and Traffic Performance Evaluation. Report #2, Oregon Transportation and Education Consortium, June 2011.

3. Quayle. S., Koonce, P., DePencier, D. and D.M. Bullock. Arterial Performance Measures with Media Access Control Readers Portland, Oregon Pilot Study. In *Transportation Research Record* No. 2192, Transportation Research Board of the National Academies, Washington, DC, 2010, pp 185-193.

Appendix A – Psuedo-code for Bluetooth Algorithms

The Bluetooth matching algorithm matches raw (Bluetooth) MAC address readings from an origination and destination pair. Multiple readings of a MAC address at a location are handled by selecting the last (most recent) of the multiple timestamps for that MAC address; this strategy is also known as last-last matching. After removing (stripping) the multiple timestamps, the algorithm compares readings from the origin/destination; a match exists if the same MAC address is read at the origin and destination within 20 minutes of each other. Once a match is found, the algorithm produces the travel time.

Bluetooth Matching Algorithm
Input *oList*: Bluetooth data for start place, *dList*: Bluetooth data for end place
Output *mList*: List of matched Bluetooth data

Execute Stripping Algorithm (:*oList)* //Remove duplicated entries in *oList*
Execute Stripping Algorithm (:*dList)* //Remove duplicated entries in *dList*

For each Bluetooth data *m* in *oList* **do**

IF mac address of *m* is found in *dList* with a timestamp after the timestamp of *m* **Then**

Insert *m* into *mlist*

End IF

End For;

Return *mlist*

End.

Filter Algorithm

Bluetooth Matching Algorithm will return a list of Bluetooth raw data that is matched from two different places for a day. The list presents flow-data with travel time between two places, but it also includes noise data that is not passed through the segment between two places. Filter Algorithm works for trying to remove the noise data. It will separate the matched Bluetooth data into expected flow-data and detoured for the segment

Procedure Filter Algorithm
Input *List*: matched Bluetooth data

For each Bluetooth data *m* from *List* **Do**

Calculate standard deviation (*sd*) and the mean (*u)* for travel time from 10 previous data from *m* in *List*

IF travel time of *m* < *u + sd* **Then**

Insert *m* into list of *good-data*

ELSE

Insert *m* into list of *bad-data*

End For;

Return *(good-data, bad-data)*

End.

Stripping Algorithm

The collected entries by DCU might be duplicated in terms of that the same MAC addresses are captured in 20 minutes time range. If there is any duplicated entry, stripping algorithm will only keep the most recent raw entry among the duplicated Bluetooth data entries.

Procedure Stripping Algorithm
Input *List*: Bluetooth raw data
Output *List*

For each *m:* Bluetooth data in *List* **do**

Find *sub-list* of Bluetooth data that have a same mac-address and a timestamp in 20 minutes after the timestamp of *m* in *List*

Remove *sub-list* From *List*

End For;

End